

Wissensmanagement mit Ontologien und Metadaten

Steffen Staab

Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB)

Universität Karlsruhe (TH)

76218 Karlsruhe

Habilitationsschrift zur Erlangung der Venia Legendi

Tag des Habilitationskolloquiums: 29 Mai 2002

Inhaltsverzeichnis

Wissensmanagement mit Ontologien und Metadaten (Übersichtsartikel)

Von S. Staab

Angenommen zur Veröffentlichung in: Informatik Spektrum, 2002

Knowledge Processes and Ontologies

Von S. Staab, H.-P. Schnurr, R. Studer, Y. Sure

Veröffentlicht in: *IEEE Intelligent Systems*, 16(1): 26-35, January/February 2001

Ontology Learning for the Semantic Web

Von A. Mädche, S. Staab

Veröffentlicht in: *IEEE Intelligent Systems*, 16(2): 72-79, March/April 2001.

Ontologies in RDF(S)

Von S. Staab, M. Erdmann, A. Mädche

Angenommen zur Veröffentlichung in: *ETAI Journal - Section on Semantic Web* (Linköping Electronic Articles in Computer and Information Science), 6, 2001.

Knowledge Portals – Ontologies at Work

Von S. Staab, A. Mädche

Veröffentlicht in: *AI Magazine*, 22(2). AAAI Press, Summer 2001, pp. 63-75.

Smart Task Support through Proactive Access to Organizational Memory

Von S. Staab, H.-P. Schnurr

Veröffentlicht in: *Knowledge-based Systems*, 3(5): 251-260. Elsevier, 2000.

Curriculum Vitae (Stand 17. Mai 2002)

Schriftenverzeichnis (Stand 17. Mai 2002)

Wissensmanagement mit Ontologien und Metadaten

Steffen Staab

Institut AIFB, Universität Karlsruhe, 76128 Karlsruhe

<http://www.aifb.uni-karlsruhe.de/WBS/sst>

<mailto:staab@aifb.uni-karlsruhe.de>

Zusammenfassung. Der Erfolg von Organisationen hängt unmittelbar von dem zur Verfügung stehenden Wissen ab. Wissensmanagement dient dazu, die Kreierung und Nutzung des Wissens nicht nur dem Zufall zu überlassen, sondern auch gezielt Verbesserungen vorzunehmen. Dabei müssen Wissensmanagementmaßnahmen primär die nicht-technologischen Aspekte berücksichtigen. Sekundär, aber eminent wichtig für effizientes Arbeiten, sind Wissensmanagementsysteme, die nicht nur eine IT-Infrastruktur bieten, sondern Wissen inhaltsbasiert zur Verfügung stellen. Für diese Aufgabenstellung eignen sich Ontologien und Metadaten, die maschinelle Ausführbarkeit mit zielgruppenorientierten, eher weichen Wissensmanagementfaktoren verbinden.

Der Kerngedanke dieser Habilitationsschrift besteht darin, einen umfassenden Rahmen für Wissensmanagementsysteme zu konstruieren, der die Vorteile von Ontologien und Metadaten nutzt. Um diesen Rahmen bereitstellen zu können, werden Arbeiten auf verschiedenen Ebenen beschrieben: 1. Es wird eine Gesamtsicht dargelegt, die die Anwendung Wissensmanagement genauso berücksichtigt wie die Besonderheiten von Ontologien und Metadaten. 2. Es wird ein Wissensmetaprozess beschrieben, der es erlaubt, Ontologien mit Hinsicht auf die Wissensmanagementanwendung zu entwickeln. Für diesen Entwicklungsprozess wurden Methoden und Werkzeuge erforscht und bereitgestellt. 3. Der Wissensprozess, der den Anwender des Wissensmanagementsystems unmittelbar betrifft, wird beschrieben und vor allem auch durch ontologie- und metadatenbasierte Werkzeuge und Systeme unterstützt.

1. Einleitung

Firmen müssen im globalen Wettbewerb zum Beispiel durch die Qualität ihrer Produkte, ihre Innovationskraft, ihre kurzen Produktentwicklungszeiten oder durch ihr erfolgreiches Management von komplexen Dienstleistungen bestehen können. In diesem Kontext wurde Wissen in den letzten Jahren als *der* zentrale Erfolgsfaktor identifiziert, der den Wert eines Unternehmens oft sehr viel stärker beeinflusst als das vorhandene Sacheigentum. Die konsequente Nutzung und Vermehrung von Wissen ist daher ein Ziel, das wesentlich zum

Unternehmenserfolg beiträgt – ein Ziel, dessen Erreichung man durch organisierte Maßnahmen, d.h. durch *Wissensmanagement*, fördern möchte.

Erfahrungen aus konkreten Wissensmanagementprojekten in Firmen haben gezeigt, dass erfolgreiches Wissensmanagement auf verschiedenen Säulen aufbauen muss. Insbesondere müssen die Beschäftigten, die eigentlichen Wissensträger, in die Maßnahmen integriert werden und sie müssen von den Maßnahmen profitieren. Ohne ihre Bereitschaft, ihr Wissen weiterzugeben, ist Wissensmanagement nicht möglich.

Die *Informationstechnologie* übernimmt beim Wissensmanagement die Rolle einer Kommunikationsplattform im allerweitesten Sinne, die zum Einsatz kommt, wenn Wissen geographisch weitläufig verteilt oder asynchron bereitgestellt werden muss (z.B. über Dokumentenmanagementsysteme). Um effektiv zu wirken, ist es nicht ausreichend, Wissen digital, aber unstrukturiert bereitzustellen. Vielmehr werden Methoden zur Strukturierung von Wissen benötigt.

Ontologien bieten sich als ein natürliches Mittel an, um Wissen strukturiert mittels Informationstechnologie bereitzustellen, ohne dabei den Human Factor zu vernachlässigen. In Anwendungen des Wissensmanagements wird Wissen häufig durch Dokumente transportiert. Die Verknüpfung von Ontologien und Faktenwissen ergibt sich durch die Benutzung verschiedener Arten von Metadaten über Dokumente sowie der Integration von sonstigen ontologiebasierten Daten.

Überblick. Im folgenden charakterisieren wir Ziele und Erfolgsfaktoren für das Wissensmanagement. Insbesondere betrachten wir zentrale „Säulen“, auf denen Wissensmanagement aufbaut (Abschnitt 2) und von denen die Informationstechnologie nur eine darstellt. Auf der Suche nach einer Charakterisierung von Wissensmanagementanwendungen beschreiben wir zunächst die kognitiven Prozesse beim Wissensmanagement in Anlehnung an Takeuchi & Nonaka [73], deren Darstellung auf der Dichotomie zwischen *implizitem* und *explizitem Wissen* beruht (Abschnitt 3). Die kognitiven Prozessschritte des Wissensmanagements lassen sich als Konvertierungen zwischen diesen beiden Wissensformen beschreiben, die durch IT-Werkzeuge unterstützt werden können. Darauf aufbauend können wir Ontologien als einen zentralen gemeinsamen Nenner verschiedenster Werkzeuge für das Wissensmanagement identifizieren. Die weitere Benutzung des Begriffs „Ontologie“ erfordert aufgrund seiner zentralen Stellung eine ausführlichere Definition (Abschnitt 4). Wir werden Beispiele zeigen für (ontologiebasierte) Metadaten, die die Menge und die Qualität des vom Rechner verarbeitbaren Wissens erweitern (Abschnitt 5). Aus dieser Beschreibung lassen sich Möglichkeiten gewinnen, die Wissensprozesse des Wissensmanagements zu unterstützen. An dieser Stelle wird es nötig, die von Takeuchi & Nonaka übernommene Beschreibung der Wissensprozesse

abzuwandeln, um zu einer neuen Prozessbeschreibung zu gelangen, die dem Blickwinkel „Wissensmanagementanwendung“ entspricht und vor allem auch den Besonderheiten von Ontologien und Metadaten besser gerecht wird (Abschnitt 6). Diese neue Beschreibung der Wissensprozesse umfasst sowohl den Ablauf für die Erstellung und Pflege von Ontologien (Abschnitt 7) als auch die darauf basierende Metadatenkreierung (Abschnitt 8). Auf die Gesamtnutzung der vorgestellten Techniken wird beispielhaft in Abschnitt 9 – Wissensnutzung – eingegangen. Vergleichbare Ansätze werden in Abschnitt 10 kontrastiert, bevor in Abschnitt 11 ein Abriss der in dieser Arbeit geleisteten Schwerpunkte erfolgt.

2. Faktoren des Wissensmanagements

Lernende Organisationen. Organisationen verfolgen Ziele gegenüber ihren Kunden, gegenüber ihren Mitarbeitern und gegenüber ihren Anteilseignern. Um diese Ziele zu erreichen, bedarf es Produktionsmittel wie Maschinen. Vor allem aber bedarf es Menschen, die die Fähigkeit besitzen, die vorhandenen Mittel zum Erreichen der Ziele einzusetzen. Um im globalen Wettbewerb um Kunden, Mitarbeiter und Investoren bestehen zu können, müssen Organisationen darüber hinaus in der Lage sein zu lernen. Das heißt, die Organisation als Ganzes muss ihre Fähigkeiten zum Erreichen ihrer Ziele verbessern können. Folgt man Peter Senge's Definition [48], „knowledge is the capacity for effective action“, ist die Nutzung und Vermehrung von *Wissen* in der Organisation der Faktor, der am besten dazu dienlich ist, den Wert einer Organisation zu erhöhen. Andere plausible Definitionen, wie zum Beispiel in [73], basieren ebenfalls auf der Zielgerichtetheit und Kontextabhängigkeit von *Wissen* im Gegensatz zur *Information* und führen damit zur selben Konklusion. Die häufig benutzte Metapher einer Organisation als einer informationsverarbeitenden Einheit erklärt zwar Standardabläufe, aber nicht die Innovationsfähigkeit von Organisationen – dazu bedarf es Wissen.

Wissensmanagement. Wissensmanagement verfolgt das Ziel, die Voraussetzungen zu schaffen bzw. zu verbessern, so dass die Mitarbeiter ihre Leistungsfähigkeit, Ziele der Organisation zu erreichen, steigern können. Erfolgreiches Wissensmanagement muss dabei als ein ganzheitliches Ziel verstanden werden. Die Einführung einer bestimmten Software oder einer bestimmten IT-Infrastruktur genügt nicht, um eine Wissensmanagementmaßnahme erfolgreich zu machen. Vielmehr müssen diverse Faktoren¹ beitragen [49]:

1. Menschen und Unternehmenskultur
2. Organisation für das Wissensmanagement

¹ Die genaue Anzahl und Benennung der Faktoren lässt sich diskutieren – ohne allerdings einschneidende Unterschiede in den Schlussfolgerungen für das Wissensmanagement zu zeitigen.

3. Wissensmanagement und Kernprozesse der Organisation
4. Wissensmanagementfokus
5. Technologien für das Wissensmanagement

Menschen und Unternehmenskultur. Wissensmanagement zielt auf das Wissen von Menschen, auf das Teilen, das Vermitteln und das Weitergeben von Wissen. Damit Wissensmanagementmaßnahmen Erfolg haben können, muss die Kultur im Unternehmen geeignet sein zur Weitergabe von Wissen. Erfolgreiche Patentrezepte sind rar, allerdings sind Kaffeeküchen als informeller Treffpunkt, Vorgesetzte, die das Mitteilen von Wissen vorleben, oder die gezielte Förderung von sozialer Kompetenz im Allgemeinen eher geeignet als eine unternehmerische Ausrichtung, die jeden Mitarbeiter zu seinem eigenen Profitcenter macht.

Organisation für das Wissensmanagement. Der Austausch, die Vermehrung und die Weitergabe von Wissen in einer Organisation geschieht teilweise durch erprobte Mechanismen, wie Aus- und Weiterbildung oder Traineeprogramme. Partiiell erfolgt sie ohne jegliches Eingreifen durch die Organisation, nämlich dadurch, dass Menschen miteinander reden oder sich gegenseitig helfen und dabei Wissen austauschen. Weitere Maßnahmen, um Wissen zu verbreiten und zu vermehren, benötigen aber in der Regel wohldefinierte organisatorische Grundlagen. Beispiele hierfür sind der *Wissensbroker*, ein Experte für bestimmte Themen, der organisationsintern Beratung leistet oder entsprechende Beratung vermitteln kann, oder *Communities of Practice*, das sind Netzwerke von Mitarbeitern zu bestimmten Themen. Auch wenn es sehr verschiedene Modelle der Ausgestaltung gibt (z.B. Finanzierung als Stabsabteilung bzw. als interner Dienstleister mit entsprechender Verrechnung), beruht der Erfolg solcher Wissensmanagementmaßnahmen in der Regel auf einer Verknüpfung von Aufgaben und Personen sowie der Unterstützung und Verantwortung durch das Top-Management – z.B. indem ein CKO, ein *Chief Knowledge Officer*, an die Geschäftsführung berichtet und seine Ziele vom Top-Management unterstützt werden.

Wissensmanagement und Kernprozesse der Organisation. Eines der Haupthemmnisse für erfolgreiches Wissensmanagement ist die Arbeitsbelastung der Wissensträger. Zusätzliche Wissensmanagementaufgaben lassen sich nur begründen, wenn der Nutzen unmittelbar sichtbar ist, außerdem werden diese nur dann wirklich umgesetzt und nicht beliebig lange Zeit verschoben, wenn sie an die Kernprozesse der Organisation (z.B. die Kundenprojekte in einem Beratungsunternehmen) gekoppelt sind. Das bedeutet, es sollte definierte Prozessschritte in den Kernprozessen der Organisation geben, an denen Wissen eingebracht, abgefragt oder bewertet wird. Zum Beispiel können Projektmeilensteine gut mit der Aufgabe verknüpft werden, Erfahrungen aus dem bisherigen Projektverlauf festzuhalten.

Wissensmanagementfokus. Eine Grundlage für erfolgreiche Wissensmanagementmaßnahmen ist die Identifikation eines oder einiger weniger Wissensmanagementfoki. Insbesondere aufwändige Maßnahmen sollten eng an Kerngeschäft und Strategie der Organisation gekoppelt sein und sich damit an den Wissenszielen der Organisation orientieren. Hilfreich für die Fokusbestimmung ist die Identifikation von Wissen [45] und seine Bewertung [30] und die Analyse von Kernprozessen und Wissensdefiziten [52].

Technologien für das Wissensmanagement. Aus der Fülle der bisher angesprochenen Themen heraus, lässt sich bereits abschätzen, dass das *eine* Softwarewerkzeug oder *die* *eine* IT-Infrastruktur, die gleichermaßen alle IT-Anforderungen für das Wissensmanagement erfüllen, nicht existieren können. Allerdings sollte man die Frage stellen, welche prinzipiellen Vorgänge ablaufen, wenn Informationstechnologie benutzt wird, um Wissensmanagement zu unterstützen. Hierfür ist es naheliegend, zunächst einmal die prinzipiell im Wissensmanagement ablaufenden Vorgänge zu beleuchten, um basierend auf deren Beschreibung weitere Schlussfolgerungen zu ziehen.

3. Kognitive Grundlagen für Wissensmanagementanwendungen

Der Ansatz von Takeuchi & Nonaka [73] scheint uns hier am ehesten geeignet, um die verschiedenen Möglichkeiten für die IT-Unterstützung in einem ersten Schritt aufzuzeigen, auch wenn man fast nie eine eindeutige Abbildung von Kategorie (wie z.B. Sozialisierung in Abbildung 1) nach Systemtyp (wie z.B. Expertenverzeichnis in Abbildung 1) oder umgekehrt finden wird (vgl. z.B. auch [28], Abb. 5).

Zentral ist bei Takeuchi & Nonaka der kognitive, auf den Menschen fokussierte Ansatz, der zwischen implizitem Wissen (genauer: „tacit knowledge“) und explizitem Wissen („explicit knowledge“) unterscheidet. Implizites Wissen ist individuell, kontext-spezifisch und daher schwer zu formalisieren und zu kommunizieren. Explizites Wissen dagegen ist kodifiziert, z.B. als Schriftstück. Die „Konvertierung“ zwischen implizitem und explizitem Wissen ist kein einfacher Vorgang, sondern ein kreativer Prozess der beteiligten Personen. Abbildung 1 zeigt die Konvertierungsmöglichkeiten nach [73], nämlich *Externalisierung*, *Kombination*, *Internalisierung* und *Sozialisierung*. Der Lernprozess von Individuen in der Organisation erfolgt in einem Wechselspiel zwischen den verschiedenen Konvertierungen. IT-Systeme, die den Lernprozess in der Organisation unterstützen sollen, müssen daher an einem oder an mehreren dieser Konvertierungsschritte direkt oder mittelbar ansetzen. Beispielhaft sind einige Möglichkeiten für IT-Systeme in das Schaubild integriert. Zu den einzelnen Konvertierungsschritten:

Sozialisierung ist die Vermittlung impliziten Wissens von Person zu Person. IT-Systeme können hierbei naturgemäß nur sehr eingeschränkt helfen. Wichtigster Nutzen einer IT-

Lösung besteht in der Regel in der Identifikation und Lokalisierung von Wissensträgern, z.B. mittels Expertenverzeichnissen im Intranet, sowie in der Errichtung einer synchronen Kommunikation, z.B. mittels Telekonferenz. Recommender Systeme können genutzt werden, um die Bewertung von Information durch Individuen oder Gruppen zu transportieren und damit die spätere Auswahl von explizitem Wissen für das Lösen einer Aufgabe zu beeinflussen.

Externalisierung. Die Kodifizierung impliziten Wissens wird als Externalisierung bezeichnet. Typische IT-Systeme für das Wissensmanagement unterstützen die Verwaltung z.B. von externalisiertem Erfahrungswissen in Datenbanken, Dokumentenmanagementsystemen oder Experience Factories [74],[7]. Flexible Werkzeuge, die diesen Prozess der Wissensakquisition basierend auf Metamodellen unterstützen, wie zum Beispiel Protégé [42], tauchen in der Wissensmanagementpraxis bisher selten auf.

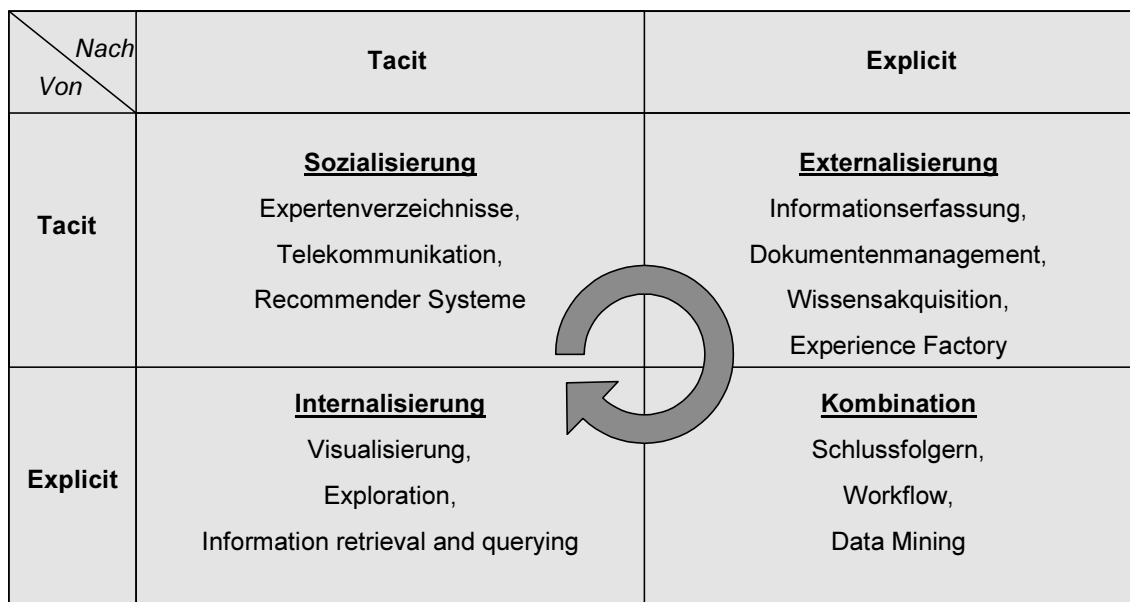


Abbildung 1: IT-Unterstützungsmöglichkeiten für die Wissenstransformierung
 (Erweiterung von [73], Abb. 3, um IT Aspekte)

Kombination. Dieser Konvertierungsschritt erfordert die Rekombination von verschiedenen Wissensbeständen (z.B. Fakten oder Kategorisierungen). An dieser Stelle können Techniken wie intelligentes Schlussfolgern (z.B. [31], [13]), Workflow [47], OLAP oder Data Mining [33] zum Einsatz kommen. Der Einsatz dieser Techniken widerspricht nicht dem Gedanken eines kreativen Vorgangs. So sind zum Beispiel Techniken wie Data Mining als Hilfsmittel zu verstehen, die nicht *per se* sondern erst im Kontext eines konkreten Data

Mining Prozesses [11] mit kreativer Fragestellung und –abarbeitung Einsichten generieren können.

Internalisierung. Die Lektüre von Dokumenten, das Nachvollziehen visueller Metaphern und Analogien und – vor allem anderen – „learning by doing“ bilden die Grundlage für die Internalisierung von explizitem Wissen. IT Systeme unterstützen hier hauptsächlich die Vorbedingungen für die Internalisierung, z.B. indem sie mittels Browsing oder Information Retrieval das Finden von explizitem Wissen ermöglichen. Darüber hinaus können aber geeignete Werkzeuge die Lektüre unterstützen oder Visualisierungen anbieten, die das Verständnis, den Internalisierungsschritt, erleichtern [28].

Gemeinsame Grundlage für Wissensmanagementanwendungen. Bei dieser Vielfalt von Möglichkeiten, die ja auch durch die verschiedenen in Abbildung 1 eingetragenen Werkzeuge und Systeme illustriert werden, stellt sich zwangsläufig die Frage: Gibt es ein Prinzip, das allen IT Systemen für das Wissensmanagement zu Grunde liegt?

Man kann das bisher Gesagte so zusammenfassen, dass IT Systeme für das Wissensmanagement Kommunikation von Wissen erlauben oder erleichtern. Dabei benutzen wir den Begriff „Kommunikation“ in einem sehr weitgefassten Sinn, der zum Beispiel auch die asynchrone Weitergabe von Wissen durch das Einpflegen von Wissensbestandteilen in eine Datenbank durch eine Person und die Rezeption durch eine zweite Person beinhaltet. Deswegen umfasst Kommunikation in dieser Deutung insbesondere die Konvertierungen zwischen implizitem und explizitem Wissen, wie sie in Abbildung 1 unterschieden werden. Dementsprechend stellt Kommunikation das Bindeglied dar zwischen Expertenverzeichnissen, die dazu dienen, Kommunikation zu etablieren, Dokumentenmanagementsystemen, die die Ablage von expliziertem Wissen erlauben, Data Mining, welches Information verdichtet, und Visualisierung als einer Modalität für die Kommunikation von Wissen. Per Definition ist dabei jegliches vom Rechner verarbeitbare Wissen explizit.

Der gemeinsame Nenner, der sich dabei – meist stillschweigend, aber dennoch präsent – durch alle IT unterstützte Kommunikationsformen hindurchzieht, sind explizit gefasste, maschinell abarbeitbare Modelle von Individuen, von Organisationen, von Prozessen und von Themen, auf die sich die Beteiligten geeinigt haben. Das sind Ontologien.

4. Ontologien

Ontologien² sind formale Modelle einer Anwendungsdomäne, die dazu dienen den Austausch und das Teilen von Wissen zu erleichtern (vgl. [21],[37]). Auf der methodischen Seite werden Techniken der objektorientierten Modellierung konsequent so weiterentwickelt, dass die Modelle nicht bloß zur Strukturierung von Software dienen, sondern auch ein explizites Element der Benutzungsschnittstelle darstellen und zur Laufzeit verwendet werden. Auf der soziokulturellen Seite erfordern Ontologien daher die Einigung einer Gruppe von Anwendern auf die jeweiligen Begriffe und deren Zusammenhänge.

Referenz und Bedeutung. Ontologien dienen der Verbesserung der Kommunikation zwischen menschlichen und maschinellen Akteuren. Hierbei befinden sich die Akteure (ob mit oder ohne Ontologie) in einer Kommunikationssituation, deren herausragende Eigenschaften durch das semiotische Dreieck [43] aufgezeigt werden.

Das semiotische Dreieck illustriert die Interaktion zwischen Worten (oder allgemeiner: Symbolen), Begriffen und realen Dingen in der Welt (vgl. Abbildung 2). Worte, die benutzt werden, um Informationen zu übertragen, können die Essenz einer Referenz, das ist der Begriff oder das referenzierte Ding in der Welt, nicht vollständig erfassen. Dennoch gibt es eine Korrespondenz zwischen Wort, Begriff und Ding.

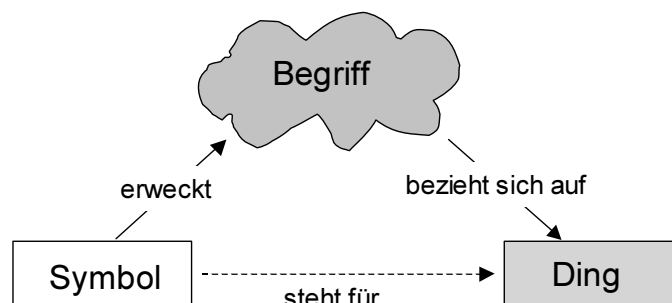


Abbildung 2: Semiotisches Dreieck

(in der Tradition von Peirce, Saussure und Frege)

Die Auswahl einer bestimmten Korrespondenz aus der Vielzahl *a priori* möglicher Korrespondenzen geschieht durch den Empfänger einer Nachricht. Hierbei benutzen

²Das Wort "Ontologie" ist zusammengesetzt aus dem Griechischen "ontos" für Sein und "logos" für Wort [54]. Es ist eine vergleichsweise neue Bezeichnung aus dem 19. Jahrhundert, die benutzt wird, um die Lehre vom Sein zu unterscheiden von der Lehre des Seienden in den Naturwissenschaften. Die ursprünglich von Aristoteles benutzte Bezeichnung war „Kategorie“ (κατηγορία). Aristoteles verwendete Kategorien, um alles, worüber gesprochen werden kann, zu klassifizieren.

verschiedene Empfänger unter Umständen verschiedene Begriffsbildungen und haben einen variierenden Erfahrungshintergrund, was wiederum zu verschiedenen Resultaten bezüglich der Korrespondenz zwischen einem Wort und den möglichen Begriffen und Dingen in der Welt führen kann.

Logik. Eine Ontologie wird ausgedrückt durch eine logische Theorie, die sich zusammensetzt aus einem Vokabular und einer Menge von logischen Aussagen zu der jeweils interessierenden Anwendungsdomäne. Die logische Theorie spezifiziert Beziehungen zwischen Worten und schränkt dabei die Menge der möglichen Interpretationen für Worte und ihren zugehörigen Beziehungen ein.

Auf diese Weise reduziert eine Ontologie die Anzahl möglicher Korrespondenzen zwischen Worten und Dingen, die der Empfänger einer Nachricht, der sich auf eine Ontologie festgelegt hat, als gültig interpretieren kann. Idealerweise bleibt im Kontext von Kommunikationssituation und Ontologie für jedes Wort aus dem Vokabular genau eine Korrespondenz mit Begriffen und Dingen in der Welt übrig.

Wissensmanagementanwendungen benutzen Verzeichnisstrukturen, Thesauri [20], Taxonomien, semantische Netze [55] oder Topic Maps [75], um Wissen abzulegen, Wissensstrukturen zu navigieren und Wissen wieder auffindbar zu machen. Ontologien fassen die verschiedenen Aspekte dieser Mechanismen zusammen. Ontologien erlauben Ableitungen (wie semantische Netze), benutzen Klassifizierungen (wie Taxonomien), beschreiben Begriffe, auf die sich eine Gruppe von Benutzern geeinigt hat (wie Thesauri), und sie dienen der Navigation in Wissensbeständen ([58]; wie Verzeichnisstrukturen und Topic Maps).³

5. Metadaten, Metainformation, Metawissen

Vom Rechner verarbeitbares Wissen ist also explizites Wissen, wobei Ontologien Modelle darstellen, die es einem erlauben, die Kommunikation von Wissen – z.B. von Mensch zu Maschine oder von Maschine zu Maschine – präziser zu fassen. Voraussetzung für die Kommunikation (im weitesten Sinne) mit der Maschine ist allerdings, dass das explizite Wissen nicht nur für den menschlichen Leser verstehbar ist, sondern dass auch der Rechner den Bezug zwischen Ontologie und Wissensbestandteilen herstellen kann.

Formalisierung. Hier gibt es zwei extreme Möglichkeiten, nämlich die vollständige Formalisierung von explizitem Wissen oder überhaupt keine Formalisierung von explizitem

³ Weitere Anwendungsbeispiele für Ontologien sind in [37], in: <http://www.ontology.org>, <http://www.semanticweb.org> oder in <http://www.ontoweb.org> skizziert.

Wissen. Das erste Extrem leidet darunter, dass in der nahen Zukunft keine Systeme absehbar sind, die die automatische Transkription von unformalen Wissen (z.B. aus Textdokumenten) in formales Wissen mit ausreichender Qualität erlauben. Aufgrund des mit der manuellen Formalisierung verbundenen Aufwandes ist dieser Weg derzeit nicht praktikabel. Sehr selten wird auch das zweite Extrem verfolgt, nämlich überhaupt keine Formalisierung, nicht einmal eine sehr simple Verzeichnisstruktur, zu benutzen, weil diese Herangehensweise relevantes Wissen nahezu unauffindbar werden lässt.

In Praxis und Forschung findet man also Mittelwege zwischen beiden Extremen, die sich gut mit dem Stichwort *Metadaten* beschreiben lassen. Metadaten sind Daten, die ausgewählte Aspekte anderer Daten beschreiben. Im allgemeinsten Fall können die Daten und/oder die Metadaten formal, semiformal oder unformal sein. Im typischen Fall einer Wissensmanagementanwendung sind die Daten unformal (z.B. Freitext) und die Metadaten semiformal, z.B. ein Dublin Core Feld für Autorennamen (<http://www.dublincore.org>), oder formal, z.B. ein Dublin Core Feld für die verwendete Sprache im Dokument, welches dann gemäß der Konvention für Sprachbezeichner [4] interpretiert werden kann. Das konkrete Format in dem die Metadaten abgelegt sind, spielt hierbei für die Ziele des Wissensmanagements eine untergeordnete Rolle. Ohnehin existieren für Metadatenstandards wie Dublin Core Realisierungen in unterschiedlichen Formaten, z.B. in XML mit DTD (<http://dublincore.org/documents/dcmes-xml/>) oder in RDF und RDF Schema (<http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/>). Es ist allerdings wichtig, eine möglichst große Übereinstimmung mit existierenden Metadatenstandards anzustreben, um eine nachhaltige Benutzung von organisationsinternen Metadaten und eine möglichst einfache Integration von organisationsexternen, „fremden“ Metadaten zu erreichen.

Ontologiebasierte Metadaten. Eine Möglichkeit, um formale Metadaten zu erhalten, ergibt sich aus der Verwendung von Ontologien bzw. ontologiebasierten Metadaten. Das Ziel sollte hierbei sein, verschiedene existierende standardisierte (ontologiebasierte) Metadaten schemata zu integrieren und um eigene Ontologien zu erweitern. Vorschläge, wie von Schreiber *et al.* [51], zeigen, wie eine modulare Verwendung mehrerer ontologiebasierter Metadatenfestlegungen bewerkstelligt werden kann. So kreieren Schreiber *et al.* Metadaten für Fotos, die auf zwei verschiedenen Ontologien beruhen. Eine Ontologie beschreibt strukturelle Daten (photo annotation ontology), z.B. wie, wo und wann wurde das Foto gemacht. Die zweite Ontologie beschreibt das Vokabular der Anwendungsdomäne (subject matter ontology), z.B. welche Gegenstände finden sich auf einem Foto. Aufgrund der festgehaltenen ontologiebasierten Metadaten lassen sich später präzise Suchanfragen stellen und beantworten.

Metainformation und Metawissen. Der Vollständigkeit halber möchten wir an dieser Stelle darauf verweisen, dass Metadaten je nach ihrem Einsatzkontext Rollen als Informationen bzw. Metainformationen sowie als Wissen bzw. Metawissen annehmen. Metadaten, z.B. über Dokumente, die kontextabhängig und zielgerichtet Aktionen beeinflussen, müssten eigentlich mit „Metawissen“ bezeichnet werden. Wir schließen uns im weiteren allerdings dem allgemeinen Sprachgebrauch an und reden weiterhin nur von Metadaten.

6. Wissensprozesse

Die Schleife über Sozialisierung, Externalisierung, Kombination und Internalisierung beschreibt den kognitiven Prozess, durch den Wissen generiert und geteilt wird. Schwerpunkte für Wissensmanagementmaßnahmen können im Prinzip aus der Analyse dieser Schritte abgeleitet werden. Z.B. kann die Notwendigkeit eines Expertenverzeichnisses abgeleitet werden, wenn das relevante Wissen kaum explizit vorliegt oder explizite Angaben zu schnell veralten.

Allerdings, so gut sich diese Prozessbeschreibung von Takeuchi & Nonaka [73] für die Beschreibung der kognitiven Abläufe im Wissensmanagement eignet – z.B. um die Wichtigkeit von Kommunikation in ihren verschiedensten Formen abzuleiten oder um die Nutzung von Wissen zu beschreiben (vgl. auch Abschnitt 9) – so schlecht lassen sich aus dieser Beschreibung Handlungsanweisungen ablesen, die den Aufbau und den Betrieb eines Wissensmanagementsystems ermöglichen.

Deswegen ist die Frage zu stellen, wie sich Wissensprozesse im Detail verhalten, die von einer Wissensmanagementanwendung unterstützt werden (sollen). Basierend auf der Zweiteilung Ontologie versus Metadaten lassen sich leicht zwei zentrale Prozesse unterscheiden, die wir hier im Folgenden anstelle der von Takeuchi & Nonaka [73] eingeführten, kognitiv ausgerichteten Prozesssicht verwenden werden. Die Kernidee unseres Vorschlags besteht darin, zwei aus Systemsicht zueinander orthogonale Dimensionen zu unterscheiden (vgl. Abb. 3).

Der **Wissensmetaprozess** umfasst alle Aspekte, die die erstmalige Erstellung der Ontologie, wie sie zur Einführung eines Wissensmanagementsystems notwendig ist, sowie ihre kontinuierliche Erweiterung und/oder Anpassung betreffen (vgl. Abschnitt 7).

Der **Wissensprozess** selbst hingegen beschreibt den Betrieb des ontologiebasierten Wissensmanagementsystems, d.h. insbesondere die Schritte zur Erzeugung und Bearbeitung von ontologiebasierten Metadaten. Aus dem Fokus „Wissensmanagementanwendung“ heraus betrachtet umfasst dieser Wissensprozess im einzelnen:

1. die Erzeugung von Wissensträgern (Dokumenten, Fakteneinträge in einer Datenbank) bzw.
2. ihrer Metadaten oder Metawissen (vgl. Abschnitt 8),
3. die Importierung externer Daten,
4. das Wiederfinden von Wissen und
5. seine anschließende Nutzung (vgl. Abschnitt 9).

Die Punkte 1 und 2 werden in Abschnitt 8 angesprochen, für den primär technischen Punkt 3 wird lediglich auf weiterführende Arbeiten verwiesen (z.B. [3],[16]), die Punkte 4 und 5 werden in Abschnitt 9 behandelt. In letztgenanntem Abschnitt wird auch noch darauf eingegangen wie man mit der Sicht von Takeuchi & Nonaka [73] die verschiedenen Nutzungsarten feiner untergliedern kann.

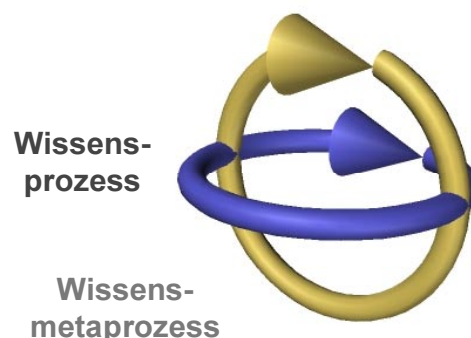


Abbildung 3: Wissensprozess und Wissensmetaprozess

7. Wissensmetaprozess

Wie bereits erwähnt umfasst der Wissensmetaprozess [65] alle Aspekte, die für die Entwicklung und die Pflege der Ontologie im Rahmen einer Wissensmanagementanwendung relevant sind. Dies beinhaltet auch, aber nicht ausschließlich, die eigentliche Ontologieentwicklung.

Unsere Vorgehensweise orientiert sich an CommonKADS [52], einer erfolgreichen Methodik für die Einführung wissensbasierter Systeme. CommonKADS umfasst auch die „weichen Aspekte“ von Wissensmanagementsystemen, z.B. die Erfassung des organisatorischen Umfeld des zu entwickelnden Systems. Allerdings stehen in CommonKADS keine Methoden zur Verfügung, die dediziert die Entwicklung von Ontologien

unterstützen. Diese Lücke schliessen wir mit unserer Beschreibung des Wissensmetaprozesses (vgl. Abbildung 4).

Dem Leser sei anzumerken, dass für eine konkrete Wissensmanagementanwendung außer dem Wissensmetaprozess und der darin enthaltenen Ontologieentwicklung (vgl. in Abbildung 4 den Ablauf von links nach rechts auf die Wissensmanagement-Anwendung hin) natürlich auch andere Entwicklungslinien relevant sind. Zum Beispiel benötigt eine Wissensmanagementanwendung graphische Benutzungsschnittstellen, Datenbanksysteme, etc., deren Entwicklungslinien hier nur angedeutet sind (in Abbildung 4 angedeutet durch Blockpfeile von rechts nach links auf die Wissensmanagement-Anwendung hin). Für diese zeitlich parallelen Entwicklungen steht aus dem Software-Engineering ein umfangreiches Methoden- und Werkzeuginventar zur Verfügung, weswegen wir uns hier auf die Aspekte der Ontologieentwicklung konzentrieren. Für die Integration der verschiedenen Entwicklungslinien bedarf es in Zukunft weiterer Forschungsarbeiten.

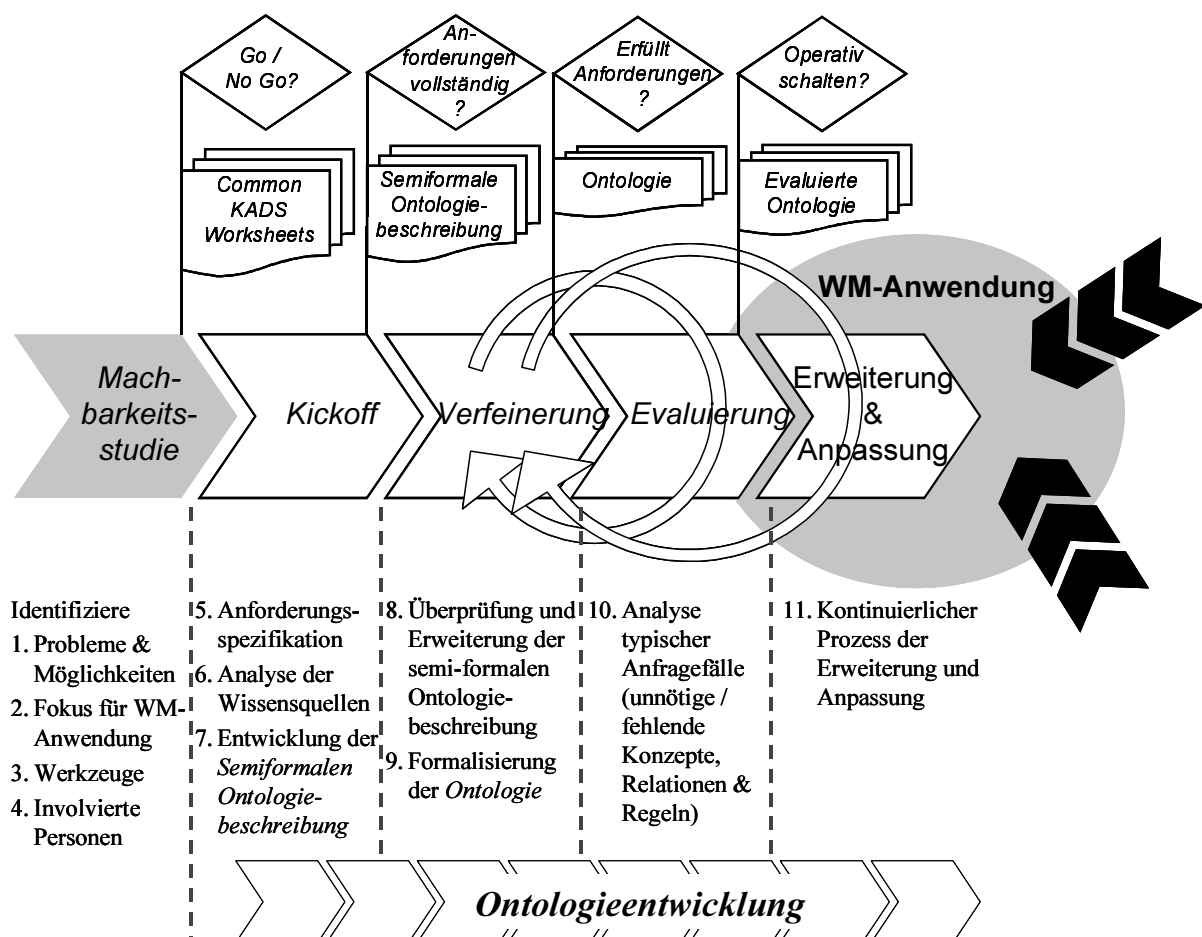


Abbildung 4: Wissensmetaprozess

Machbarkeitsstudie. Die Machbarkeitsstudie ist dem eigentlichen Ontologieentwicklungsprozess vorgelagert. In der Machbarkeitsstudie werden im Wesentlichen die organisatorischen Rahmenbedingungen von Wissensmanagementanwendungen erfasst. Dazu gehört insbesondere die Identifikation von Problemen und Möglichkeiten – wo lässt sich ein Quick Win erzielen? – woraus ein Fokus für die Wissensmanagementanwendung abgeleitet werden kann. Es stellen sich Fragen nach benutzten Werkzeugen, sowie nach den involvierten Personen. Aus der Konsolidierung dieser Information und der Integration des Managements ergibt sich eine fundierte Entscheidung für oder gegen das Projekt. Im positiven Fall fließen die bereits erhobenen Daten in die nächste Phase ein.

CommonKADS stellt für diese Phase eine Menge von Schablonen (CommonKADS Worksheets) bereit, um die Beantwortung und Beachtung der gerade skizzierten Standardfragen abzusichern. Diese Schablonen sind nicht spezifisch für die Ontologieentwicklung, vielmehr erwarten wir, dass ihre Analyse auch für parallele Entwicklungslinien (wie z.B. für die Benutzungsschnittstelle) von Interesse sein wird.

Am Ende der Machbarkeitsstudie muss „GO“ oder „No GO“ entschieden werden, d.h. ob eine ontologiebasierte Wissensmanagementanwendung zur Lösung der betrachteten Probleme angestrebt wird oder ob andere Maßnahmen zur Erreichung der Ziele der Organisation geeigneter sind.

Kickoff. Die Kickoff-Phase dient im Wesentlichen dazu, eine erste Vorstellung von der Ontologie zu gewinnen: Welche Anforderungen muss sie erfüllen, auf welchen Wissensquellen sollte sie aufbauen und wie könnte eine initiale semiformale Ontologiebeschreibung beschaffen sein?

Die Ontologiebeschreibung beruht in dieser Phase typischerweise nicht auf einer formalen, logischen Sprache, sondern auf einer Kombination von Text und un- oder nur teilweise getypten Graphen, z.B. in Form einer Mindmap.

In der Kickoff-Phase gibt es reichhaltige Möglichkeiten, die Generierung dieser semiformalen Ontologiebeschreibung zu unterstützen:

- a. *Verwaltung der Anforderungsspezifikation:* Wie im Software-Engineering werden gerne Kompetenzfragen erhoben, die die spätere Benutzung der Ontologie illustrieren. Die Verwaltung der Spezifikation erlaubt später die vereinfachte Überprüfung [35].
- b. *Verknüpfung der Anforderungsspezifikation mit der Ontologie:* Anhand der Kompetenzfragen lassen sich unmittelbar relevante Begriffe und Relationen identifizieren [71].

- c. *Analyse der Wissensquellen*. Aus den Wissensquellen lassen sich halbautomatisch, z.B. mittels Machine Learning, relevante Begriffe und Relationen erkennen (vgl. z.B. [38],[39]), die für die Ontologiebeschreibung genutzt werden.

Am Ende der Kickoff-Phase muss der Anwender/Domänenexperte entscheiden, ob die betrachteten Anforderungen vollständig sind und die relevanten Wissensquellen ausreichend abgedeckt wurden. Falls dies nicht der Fall ist, muss vor der Beendigung dieser Phase eine Nachbesserung erfolgen, ansonsten kann die Ontologieentwicklung mit der Verfeinerungsphase fortgesetzt werden.

Verfeinerung. In der Verfeinerungsphase wird zunächst die semiformale Ontologiebeschreibung auf ihre Konsistenz und Vollständigkeit überprüft und, wo nötig, erweitert. Ebenso wie die Erhebung der organisatorischen Rahmenbedingungen und der Anforderungen geschieht dies im Dialog mit dem Anwender/Domänenexperten des Systems. Die daraus resultierende Ontologiebeschreibung muss so umfassend sein, dass alle wichtigen Begriffe, Relationen und Regeln ausführlich dargelegt sind, in der Regel allerdings nicht in einer formalen Sprache. Als hilfreich – wenn auch unvollständig für die anvisierten Ziele – hat sich zum Beispiel die UML-„Standard“-Notation erwiesen.

Schließlich erfolgt in der Verfeinerungsphase die Formalisierung der Ontologie. Dabei stellen sich vor allem folgende Fragen:

- a. *Ontologiesprache*: Welche Ontologiesprache soll Verwendung finden? Aufgrund obiger Erwägungen zur Nachhaltigkeit und Integrierbarkeit anderer Module sind Standards wie z.B. RDF Schema oder DAML+OIL vorteilhaft [60],[14]. Aufgrund der derzeit fehlenden Standardisierung von Regeln finden aber häufig auch andere Sprachen [31] oder deren Kombination mit RDF Schema Verwendung [13].
- b. *Axiome*: Welche Axiome sollen Verwendung finden? Erleichterung bei der Modularisierung von Ontologiesprache und den zu verwendeten Strukturen sowie bei der Interaktion mit den Domänenexperten liefern *Patterns* [10], [61].

Am Ende der Verfeinerungsphase muss überprüft werden, ob die Ontologie die Anforderungen, die in der Kickoff-Phase erhoben wurden, erfüllt. Falls dies nicht der Fall ist, muss die Ontologie berichtigt oder erweitert werden. Im Idealfall hat man an dieser Stelle die endgültige Ontologie, deren Funktionalität und Richtigkeit noch anhand von Systemtests in der Evaluierung bestätigt werden muss.

Evaluierung. Für die Evaluierung der Ontologie in der konkreten Anwendung wird ein lauffähiger Prototyp des Wissensmanagementsystems benötigt. Anhand dieses Prototyps können typische Anfragefälle analysiert werden, um unnötige oder fehlende Konzepte, Relationen und Regeln zu identifizieren oder sogar fehlerhafte Ontologiestrukturen zu finden.

Falls sich aus der Evaluierung heraus ein Revisionsbedarf für die Ontologie ergibt, ist es nötig in die Verfeinerungsphase zurückzukehren, um anhand weiterer Diskussionen und Analysen Ontologiestrukturen zu ergänzen, zu entfernen oder zu korrigieren (vgl. erster Rückkopplungspfeil von Evaluierung nach Verfeinerung in Abbildung 4).

Erweiterung und Anpassung. Im Bereich der Erweiterung und Anpassung⁴ von Ontologien gibt es derzeit nur wenige Erfahrungen – vor allem was nichtmonotone Änderungen an der Ontologie betrifft (z.B. Löschen von noch benutzten Begriffen). Größere Änderungen, vor allem Umstrukturierungen, werden einer neuerlichen Verfeinerungs- und Evaluierungsphase bedürfen (vgl. zweiter Rückkopplungspfeil von Erweiterung und Anpassung nach Verfeinerung in Abbildung 4). Während kleinere Änderungen, z.B. Hinzufügen weniger neuer spezifischer Konzepte, den Betrieb des Systems nicht wesentlich beeinflussen werden.

Viele weitere offene Fragestellungen betreffen die Ontologieerstellung. Insbesondere existiert derzeit kein Werkzeug, welches den ganzen Wissensmetaprozess lückenlos unterstützen würde, um so die Lücke zwischen vagen Vorstellungen über Eigenschaften der Ontologie und ihrer konkreten Realisierung zu schließen.⁵ Die Erkenntnisse aus der Entwicklung und der Anwendung eines solchen Werkzeugs würden sicherlich auch auf den Prozess selbst zurückwirken. Insbesondere die Detaillierung einzelner Schritte, wie zum Beispiel mit Hilfe eines Brainstorming-Werkzeugs zur Erstellung der Ontologiebeschreibung, könnte noch leichte Verschiebungen von Resultaten und Arbeitsschritten nach sich ziehen.

8. Wissensprozess: (Meta-)Datenkreierung

Der Schritt der (Meta-)Daten-/ (Meta-)Wissenskreierung ist vorwiegend eine Mischung aus Externalisierung (z.B. der Beschreibung einer Projektidee) und Kombination (z.B. der Einordnung von Teilbeschreibungen in vorgegebene Schemata). Insbesondere der kreative Arbeitsanteil darf dabei nicht durch unnötige oder nur selten gewinnbringende Zusatzarbeit belastet werden, die dazu führt dass die Wissensmanagementumgebung umgangen wird. Hierfür gibt es aber eine Reihe von Methoden – typischerweise mit einem Trade-off zwischen Genauigkeit und Automatisierung.

⁴ Diese Phase wird häufig mit „Wartung“ oder „Instandhaltung“ bezeichnet. Beide Bezeichnungen werden aber der Aufgabe nicht gerecht. Es genügt nicht die Software mit etwaigen Schnittstellenänderungen zu versehen, vielmehr ist regelmäßiger Aufwand für die Aktualisierung der Ontologie einzuplanen.

⁵ Zu den am weitesten entwickelten Werkzeugen gehören z.B. OntoEdit™ (http://www.ontoprise.de/technology_downloads.htm) und Protégé (<http://protege.stanford.edu/download.html>).

Automatische Verfahren. Zu den traditionellen Verfahren der Metadatengenerierung gehören automatische Klassifikation [29], Informationsextraktion [6], oder auch automatisches Dokumentenclustering [53].

Die Zusammenführung von effizienten automatischen Verfahren mit Ontologien als semantischem Grundgerüst wurde bisher allerdings noch vergleichsweise selten und teilweise gar nicht durchgeführt, obwohl diese Vorgehensweise gerade für die hier dargelegten Wissensmanagementanwendungen praktisch ist, da sie auf eine einfache Integration hinausläuft. Deswegen haben wir in eigenen Arbeiten [56],[32],[59] ontologiebasierte Informationsextraktion betrachtet und ontologiebasiertes Dokumentenclustering [26], [27] eingesetzt.

Manuelle Verfahren. Typische Metadaten, die per Hand vergeben werden, sind Klassifikationslabel (Verschlagwortung) oder Autorenangaben wie in Dublin Core. Neueren Datums findet man relationale Metadaten, das sind Metadaten mit Querbeziehungen zwischen angegebenen Objekten. Relationale Metadaten können den logischen Zusammenhang zwischen Dokumenten und Dokumentinhalten beschreiben. Zum Beispiel kann man mit relationalen Metadaten angeben, dass „Dokument A erweitert das Spezifikationsdokument B um die Menge von Attributbeschreibungen M“ – woraufhin der Unterschied zwischen A und B abgefragt und mit anderen Unterschieden verglichen werden kann [24].

Bei den manuellen Verfahren sei angemerkt, dass die erwartete hohe Genauigkeit nicht immer gegeben ist. Untersuchungen im Bereich der Kategorisierung in den Informationswissenschaften haben gezeigt, dass die Übereinstimmung von menschlichen Annotierern oft erschreckend gering ist. Vorläufige Untersuchungen zur Kreierung von Metadaten mit unserem Werkzeug haben diese Aussage bestätigt und indizieren darüber hinaus, dass die Übereinstimmung für relationale Metadaten tendenziell noch schwächer ausfällt als für Kategorisierungen für Dokumente [17], [34], [62].

Semi-automatische Verfahren. Ein relativ regelmäßig benutzter Mechanismus für die Kreierung von Metadaten ist die Bereitstellung von Schablonen, z.B. XML Schemata, die mittels eines XML Editors die Eingabe von Fakten erlauben. Die Verknüpfung der XML Elemente mit konkreten Ontologiekonzepten [19] erlaubt hinterher die Weiterverwendung von präzisen Attributen ohne dass ein Aufwand durch das manuelle Annotieren nötig ist [64].

Ähnlich elegant stellt sich auch die Integration der Metadatenkreierung in den Prozess der Dokumenterstellung selbst dar [25]. Durch die Integration wird der Mehrfachaufwand vermieden, der typischerweise bei Annotierungsarbeiten entsteht, die der Dokumenterstellung nachgelagert sind.

Eher in den Bereich der Forschung fällt derzeit der Vorschlag von semantischen Annotierungen durch Informationsextraktion. Für einfache semantische Auszeichnung, nämlich für die Kategorisierung von Phrasen, existieren zwar seit einiger Zeit Systeme wie Alembic [12], für relationale Metadaten und Informationsextraktion hingegen gibt es zwar erste Konzepte [18], allerdings noch keine stringente Aufgabenteilung zwischen Annotierungswerkzeug und Informationsextraktion. Diese Aufgabenteilung ist in der Praxis aber sehr wichtig, da selbst die besten Informationsextraktionsverfahren viele Fehler produzieren.

9. Wissensprozess: Wissensnutzung

Das Prozessmodell von Takeuchi & Nonaka [73] fokussiert auf die kognitiven Vorgänge der Anwender im Wissensmanagement. Damit ist, wie bereits erwähnt, das Modell kaum geeignet, um die Einführung einer Wissensmanagementanwendung zu begleiten. Ebenso lässt sich der Lebenszyklus von Wissensbestandteilen in einer Wissensmanagementanwendung nicht gut durch das kognitive Modell erklären. Was aber die Nutzung der Wissensmanagementanwendung betrifft, kann man aufgrund der verschiedenen Interaktionen der Benutzer eine Schwerpunktsetzung gemäß der Kategorisierung in Kombination, Sozialisierung, Internalisierung und Externalisierung vornehmen. Speziell am Institut AIFB wurden verschiedene Wissensportale [63] entwickelt, die Ontologien und Metadaten benutzen, die solche verschiedenen Schwerpunktsetzungen illustrieren.

Kombination. Der Fokus des Corporate History Analyzer [5] liegt in der Kombination von annotierten Daten. Der Corporate History Analyzer vermittelt seinem Benutzer einen Überblick über Firmen in der Chemieindustrie. Hierfür wurde eine anwendungsspezifische Ontologie entwickelt, deren Besonderheit darin liegt, dass sie zeitliches Wissen berücksichtigt. Metadaten formalisieren Aktionen wie Verkäufe, Merger, Ankäufe oder Reorganisation von Firmen, wie sie in Nachrichtenmeldungen enthalten sind. Aus einer Anfangssituation heraus integriert der Corporate History Analyzer die einzelnen Bruchstücke und liefert aktuelle Strukturinformationen über Firmen. Zum Beispiel entsteht durch einen Merger aus zwei Firmen A und B eine neue Firma C. Wenn der Benutzer nach einer Übersicht zu A zum Zeitpunkt t fragt, bekommt er in Abhängigkeit von t die entsprechende Auskunft oder einen Verweis auf die neue Firma. Dabei kann der Corporate History Analyzer viele einzelne Informationsbruchstücke aggregieren, um ein vollständiges Bild zu liefern. Unter anderem kann er Vorschläge über mögliche strategische Aktivitäten liefern. Zum Beispiel könnte eine US Firma eine deutsche Firma kaufen, um in Europa zu expandieren. Der Schwerpunkt dieser Anwendung liegt in der Kombination von Einzelwissen mittels temporalem Schlussfolgern [57].

Internalisierung. Eine Vorbedingung für Internalisierung ist die zielgruppengerechte Aufbereitung von Wissen. Ein *Community Web Portal* dient als Anlaufstelle für eine Interessensgruppe, über die sie ihre Interessen formulieren und mit der sie Wissen auf dem WWW teilen kann. Am Institut AIFB haben wir basierend auf der KA2-Initiative [8], die Informationen semantisch annotiert bereitgestellt hat, ein Community Web Portal für die Knowledge Acquisition Community entwickelt [58]. Dieses KA2-Portal (<http://ka2portal.aifb.uni-karlsruhe.de/>) erlaubt vor allem die leichte Navigation mit semantischen Methoden und semantische Suche. Während das KA2-Portal eine erste Studie war, wurden die dort verwandten Techniken weiterentwickelt zu einem System im Dauerbetrieb, dem Internetauftritt des AIFB [68]. Hierzu wurden insbesondere auch einzelne Komponenten weiterentwickelt, wie zum Beispiel das semantische Ranking oder das Crawlen von RDF Fakten.

Sozialisierung. Wie oben bereits angeführt ist der unmittelbare Kontakt häufig die effektivste Art Wissen zu teilen. Expertenverzeichnisse oder – etwas allgemeiner – Skill Management Systeme haben die Aufgabe explizites Wissen über mögliche Ansprechpartner bereitzustellen. In [70] wird eine ontologiebasierte Skill Management Anwendung vorgestellt, die sowohl geeignet ist, Experten zu lokalisieren, als auch Bewerbungen mit offenen Stellen zu matchen (für Matchmaking vgl. [69]).

Externalisierung. Der Schritt der Externalisierung wird in den vorgestellten Portalen über die Kreierung von Metadaten, wie oben beschrieben, unterstützt, allerdings seltener über die Webapplikation selbst. Ausnahmen hiervon werden aber z.B. in [26] oder in [64] beschrieben. In letzterem Fall wird ein Assistenzsystem in den Wissenskreierungsprozess integriert, der die Entwicklungen eines Projektplans durch Proaktion des Systems unterstützt.

10. Einordnung verwandter Arbeiten

Bedarfsgerechte Bereitstellung und Austausch von Wissen ist das zentrale Anliegen von Wissensmanagementsystemen. Um diese Funktion zu erfüllen benötigt man

- a. eine geeignete *Infrastruktur*, die Kommunikation auf technischer Ebene, d.h. möglichst unabhängig von Ort und Zeit, ermöglicht,
- b. Komponenten, die *organisatorische Anforderungen*, wie z.B. nach Privatheit von Informationen oder nach der Eigenständigkeit von Wissensgruppen erfüllen, und
- c. *inhaltsorientierte Methoden* für den Austausch und die Weitergabe von Wissen.

Konventionelle Wissensmanagementsysteme, die auf dem Softwaremarkt zu finden sind, stellen vorwiegend Funktionalitäten für eine gute Infrastruktur (z.B. Intranet) und für organisatorische Abläufe und Kontrollfunktionen bereit (z.B. durch Content Management

Systeme). Die zentrale Frage danach, welche Inhalte in einem Wissensmanagementsystem bereitstehen oder wie diese zu verknüpfen sind wird seltener beantwortet. Oft wird nur auf Stichwortsuche verwiesen oder auf die Struktur der Graphischen Benutzeroberfläche, z.B. des Intranetzugangs zum Wissensmanagementsystem.

Reine Stichwortsuche aber überlässt das Problem des Erkennens von relevantem Wissen fast vollständig dem Benutzer. Die Bereitstellung von Inhalten allein über bestimmte Strukturen von Benutzeroberflächen wiederum widerspricht gutem Software-Design, das eher versucht „Model“, „View“ und „Control“ voneinander zu separieren, um Anpassungen leichter durchführen zu können. Ontologien stellen ein Modell zur Verfügung, das sehr komfortabel für die Konstruktion von Benutzeroberflächen als auch für die Strukturierung der Inhalte nutzbar ist, dabei aber ein höheres Maß an Modularität erreicht. Dennoch finden sich auf dem Softwaremarkt bisher am ehesten Stichwortsuche und programmierte Webseiten. Einige Firmen setzen in ihren Produkten aber auch seit geraumer Zeit z.B. Thesauri und seit neuerer Zeit Ontologien erfolgreich ein.

Auf wissenschaftlicher Seite lassen sich aufgrund der Breite der Wissensmanagementprozesse enorm viele Arbeiten zitieren, die Teilaspekte behandeln. In dieser Übersicht können wir nur wenige ontologiebasierte Gesamtkonzepte kontrastieren.

A. Rabarijaona *et al.* [46] basieren ihr Organizational Memory auf der Basis von Ontologien und XML-Dokumenten. Grundlage für ihren Ansatz bildet das Osirix-System, welches inhaltsorientierte, semantische Suche auf der Grundlage von Ontologien erlaubt.

U. Reimer, A. Margelisch und M. Staudt [47] haben in ihren Arbeiten als eine der ersten die Unterstützung von Organizational Memories mittels Ontologie- und Prozessmodellierung propagiert.

A. Abecker, A. Bernardi, A. Dengel, L. van Elst, K. Hinkelmann, O. Kühn, H. Maus, M. Sintek und S. Schwarz [1], [15] haben in ihren Arbeiten eines der umfassendsten Konzepte für Wissensmanagement mit Ontologien und Metadaten vorgelegt. Dieses umfasst die Modularisierung von Ontologien in Information, Enterprise und Domain Ontologies, was die Wartung des Gesamtsystems und seine Integration, z.B. mit Digital Libraries, erleichtert. Es beinhaltet auch eine dedizierte Integration von kontextspezifischen Diensten und Sichten, z.B. die kontextabhängige Bearbeitung von Papierdokumenten mittels OCR.

Diese Arbeiten konnten allerdings kein umfangreiches Methoden- und Technikrepertoire zum Aufbau und Betrieb ihres Systems vorweisen, wie es in diesem Übersichtsartikel hier vorgestellt wurde. Ein solches – selbst noch verbesserungswürdiges – Repertoire ist aber nötig, um mittelfristig zu praxisreifen, umfassenden Wissensmanagementsystemen zu gelangen, die die Vorteile von Ontologien und Metadaten vollständig realisieren können.

11. Einordnung eigener Arbeiten

Die für das Habilitationsverfahren beigelegten Veröffentlichungen stellen einen Querschnitt aus dem Spektrum von Arbeiten des Autors dar, deren Kontext hier beschrieben wurde. Wo angebracht, wurden die beigelegten Arbeiten bereits referenziert, ebenso wie die weiteren im unmittelbaren Zusammenhang stehenden wissenschaftlichen Veröffentlichungen.

Die hier vorgelegten Arbeiten zielen darauf ab, besonders große, hervorstechende Lücken im Bereich Wissensmanagement mit Ontologien und Metadaten zu schließen:

1. Zu diesen Lücken gehört an vorderster Stelle eine übergeordnete Prozesssicht auf das Problem, die es erlaubt, Handlungsanweisungen aufgrund von Prozessmodellen zu formulieren. Diese Lücke wurde mit dem Beitrag „**Knowledge Processes and Ontologies**“ angegangen [66]⁶. Dieser Beitrag beschreibt die zweiteilige Rolle die der Wissenszyklus bei der Einführung eines Wissensmanagementsystems innehat. Wie auch in dieser Übersicht vorgestellt, wird unterschieden in den Metawissensprozess und den „eigentlichen“ Wissensprozess. Beide Prozesse werden beschrieben und eine mögliche Abarbeitung wird anhand des Corporate History Analyzer [5] illustriert. Vergleichsweise einzigartig stehen für die Unterstützung dieser Prozesse verschiedenste Werkzeuge zur Verfügung.

In dieser übergeordneten Sicht wurden im Wissensmetaprozess verschiedene Aufgaben identifiziert. Die zur Machbarkeitsstudie zugehörigen Aufgaben (vgl. Abbildung 4) werden dabei bereits durch die CommonKADS Worksheets abgedeckt. In der Kickoff-Phase unterscheiden wir drei Aufgaben. Für die Aufgabe 5. Anforderungsspezifikation konnten wir im wesentlichen auf den Arbeiten von [35] aufbauen. Für die Aufgaben 6. Analyse der Wissensquellen und 7. Semiformale Ontologiebeschreibung allerdings ließen sich keine Grundlagen finden. Deswegen war ein wichtiger Punkt unserer Arbeiten:

2. Die (semi-)automatische Generierung einer Ontologiebeschreibung aus der Analyse von Wissensquellen, die in „**Ontology Learning for the Semantic Web**“ [39] beschrieben wird. In dieser Veröffentlichung wird ein zyklischer Analyseprozess beschrieben, der mittels verschiedener, teilweise existierender Algorithmen aus dem maschinellen Lernen unterstützt wird. Unter diesen Lernalgorithmen findet sich ein eigener Entwurf eines Algorithmus, der aufbauend auf verrauschten Daten nichttaxonomische Relationen zwischen Begriffen vorschlägt. Dieser Algorithmus wurde anhand von Testdaten eingesetzt und evaluiert [38]. Neben dem Problem

⁶ Eine Langfassung von [65] wird in [66] beschrieben.

der algorithmischen Spezifikation trat an dieser Stelle vor allem das Problem fehlender Evaluierungsstandards bei der Konstruktion von Ontologien, der zum Beispiel eine Verallgemeinerung von existierenden Vorschlägen (vgl. [22]) nötig machte.

In der Verfeinerung wird in Aufgabe 8. die Ontologiebeschreibung überprüft und gegebenenfalls erweitert. Dies ist im wesentlichen ein Dialog zwischen Ontologieentwickler und Anwender/Domänenexperten, der zwar nicht trivial ist, aber weniger Forschungsbedarf nötig hatte. Mehr Bedarf an neuen Vorschlägen war nötig im Bereich der Aufgabe 9. Formalisierung der Ontologie, da existierende Mechanismen, die es erlauben, Ontologien problemgerecht und losgelöst von ihrer Implementierung in einer konkreten Zielsprache zu formalisieren, nicht genügend ausdrucksstark waren. Diese Problematik wurde in

3. „**Ontologies in RDF(S)**“ [60] aufgegriffen. Gegenstand des vorgeschlagenen Ansatzes ist die Nutzung von einfachen Standards (hier RDF und RDF Schema) mit dem Ziel einer möglichst einfachen, interoperablen Nutzung in verschiedenen Systemen. Gleichzeitig wird durch die beschriebene Lösung, nämlich eine Objektifizierung von Axiomen, eine größtmögliche Transparenz bei der Konstruktion von Ontologien in Ontologieeditoren erreicht – ein bis dahin kaum angegangenes, aber vordringliches Problem. Als Beispiel für den Ansatz wird eine komplexe Axiomatik betrachtet, nämlich das Schlussfolgern über anatomische Relationen, wie es in [23] beschrieben ist. Unsere neueren Arbeiten greifen diese Thematik auf und entwickeln sie weiter in Richtung einer semi-formalen Spezifikation von „Semantischen Mustern“ [61]. Parallele Arbeiten, z.B. in der Digital Library Community, zeigen den allgemeinen Bedarf nach solchen Spezifikationsformen, der über die Wissensmanagementanwendung weit hinaus geht (vgl. [44]).

Des weiteren wurden in der übergeordneten Sicht verschiedene Schritte für den eigentlichen Wissensprozess beschrieben. Hier ergab sich die Anforderung, vor dem technischen Hintergrund von Ontologien und Metadaten praktikable Wissensmanagementsysteme bereitzustellen, die den Wissensprozess in einem verteilten Szenario zu unterstützen, wie es typischerweise in Unternehmen, die Wissensmanagement einsetzen, anzutreffen ist.

4. Mit diesem großen Bedarf befasst sich der Beitrag „**Knowledge Portals**“ [63]. Er behandelt die Struktur von ontologiebasierten Wissensportalen und darauf basierenden Diensten. Beispiele sind das KA2-Portal (vgl. auch [58]), ein Wissensportal von und für die Knowledge Acquisition Community. Aufbauend auf der in der KA2-Initiative [8] erarbeiteten Wissensgrundlage, nämlich Wissen über Wissenschaftler aus der Knowledge Acquisition Community, wurde ein Gesamtkonzept entworfen, das weit über die ursprüngliche Funktionalität des

Suchens und Abfragens hinausgeht und zum Beispiel ontologiebasierte Abfragemasken, ontologiebasiertes Browsing und Personalisierung enthält. Im Artikel wird bereits eine weitere Anwendung skizziert, das TIME2Research Portal, welches es Finanzanalysten erlaubt Wissen zu teilen und kontextgerecht abzugreifen.

Schließlich ergibt sich aber in allen diesen aufgeführten Prozessen und Prozessschritten die Frage, welche Aktionen vom Menschen und welche von der Maschine initiiert werden.

5. Der vielseitige Assistent als eine Metapher für ein proaktives Wissensmanagementsystem wird in „**Smart Task Support through Proactive Access to Organizational Memory**“ [64] so entwickelt, dass Wissen nicht nur kontextabhängig und reaktiv, sondern auch proaktiv bereitgestellt werden kann. Anhand eines Projektplanungsproblems wird illustriert wie durch die Modellierung von Kontext ein Assistenzsystem in die Lage versetzt werden kann, den Benutzer bei der Kombination von Faktenwissen zu unterstützen, ohne ihn zu bestimmten Abarbeitungsinteraktionen zu zwingen. Eine besonders interessante Entwicklung ergab sich aus dieser Arbeit in neuerer Zeit: In einem Projekt zur Entwicklung von Flugbuchungssystemen soll ein regelbasiertes System zur besseren Unterstützung eines naiven Benutzers bei Internetanfragen eingesetzt werden. Proaktive Assistenz wie sie in [64] vorgestellt wurde, erscheint dabei als ein vielversprechender Kandidat, um dem Benutzer langwierige und letztendlich erfolglose Buchungsanfragen zu ersparen ohne ihn dabei zu bevormunden.

Bemerkungen zu Koauthorschaft. Wie das auf dem Gebiet der angewandten Informatik häufig der Fall ist, bestehen viele der hier vorgestellten Arbeiten aus der Konzeption, der Realisierung und der wissenschaftlichen Beschreibung von Systemen. Diese Vorgehensweise ist nur selten im Rahmen einer Einzelarbeit denkbar. Der Autor hat für das Habilitationsverfahren Arbeiten vorgelegt, deren Kerngedanken von ihm entwickelt wurden, deren Umsetzung aber natürlich auch von der Beteiligung seiner Koautoren profitiert hat. Die Reihenfolge der Autoren spiegelt dabei primär die Arbeitsanteile an der jeweiligen Veröffentlichung wieder, d.h. dass die Arbeiten [65], [63], [64], und [60] primär vom Autor initiiert und durchgeführt wurden. Die Arbeit [39] lässt sich in etwa hälftig den beiden Autoren zurechnen.

12. Zusammenfassung

In diesem Artikel wurde die Zielsetzung von Wissensmanagement in Unternehmen definiert und ihre Problematik erläutert. Typische Wissensmanagementprobleme sind primär weicher Art – z.B. dass Mitarbeiter Wissen teilen statt horten. Dennoch können

Wissensmanagementsysteme erheblich zur Akzeptanz von Wissensmanagementmaßnahmen beitragen, wenn sie den Betroffenen das Leben erleichtern und nicht erschweren. Zentral ist unserer Ansicht nach die Definition von Wissensmanagementsystemen als Plattformen, die es erlauben leichter synchron oder asynchron Wissen auszutauschen. Ontologien als semantische Modelle der Anwendungsdomäne dienen in diesem Kontext als Vehikel, um Wissen präziser zwischen den Beteiligten und zwischen Mensch und Rechner auszutauschen.

Beim Aufbau eines ontologiebasierten Wissensmanagementsystems unterscheiden wir zwei Kernprozesse, nämlich den Wissensmetaprozess und den eigentlichen Wissensprozess, die verschiedene Charakteristika aufweisen. Für beide Prozesse haben wir Schritte beschrieben, die wiederum einzeln oder in Kombination (vgl. [36]) durch Methoden und/oder Werkzeuge unterstützt werden können.

Das Wissensmanagementproblem gestaltet sich im Praxisalltag allerdings vielfältiger als es hier – oder in einer anderen einzelnen Resource – beschrieben werden kann. Zum Beispiel sind Problematiken wie Wissensmanagement und eLearning eng verknüpft und werden in der betrieblichen Praxis inzwischen auch regelmäßig zusammen und mit ähnlichen Methoden angegangen (vgl. [41],[67]). Auch die für Wissensmanagementsysteme gültigen Methoden und Techniken sind viel weiter generalisierbar. Insbesondere im Rahmen des „Semantic Web“ (<http://www.semanticweb.org>) befruchten sich Techniken für Wissensmanagementsysteme und das Semantic Web gegenseitig und bieten Lösungen für vielerlei Aufgaben (vgl. z.B. [40],[67]).

Danksagung

Ich bedanke mich sehr herzlich bei allen meinen Kollegen am Institut AIFB, in der Forschungsgruppe Wissensmanagement am FZI (Forschungszentrum Informatik) und bei Ontoprise GmbH für viele Diskussionen und Anregungen. Mein besonderer Dank für Inspiration und fruchtbare Zusammenarbeit gilt Jürgen Angele, Stefan Decker, Michael Erdmann, Siegfried Handschuh, Alexander Mädche, Hans-Peter Schnurr und York Sure. Besonderen Dank möchte ich Herrn Prof. Dr. Rudi Studer aussprechen, der mir alle Freiheiten ließ, um meine Ideen zu verwirklichen, dabei aber auch immer als Ansprechpartner und Förderer präsent war.

Literatur

- [1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, M. Sintek. Toward a Technology for Organizational Memories. *IEEE Intelligent Systems*, 13(3): 40-48 (1998).

- [2] A. Abecker, S. Decker, O. Kühn. Organizational Memory - Das aktuelle Schlagwort. *Informatik Spektrum* 21(4): 213-214, 1998.
- [3] R. Agrawal, R. Srikant. On Integrating Catalogs. In: *Proceedings of the Tenth International World Wide Web Conference – WWW 10*. Hong Kong, China, May 1-5, 2001. ACM Press, pp. 603-612.
- [4] H. Alvestrand. Tags for the Identification of Languages. Internet Engineering Task Force (IETF) RFC 3066. <ftp://ftp.isi.edu/in-notes/rfc3066.txt> Abgerufen am 4. September 2001.
- [5] J. Angele, H.-P. Schnurr, S. Staab, R. Studer. The Times They Are A-Changin' - The Corporate History Analyzer. In: D. Mahling & U. Reimer. *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management*. Basel, Switzerland, October 30-31, 2000, pp. 1.1-1.11.
- [6] D. Appelt, J. Hobbs, J. Bear, D. Israel, M. Tyson. FASTUS: A Finite-state Processor for Information Extraction from Real-world Text. *IJCAI-93 - Proceedings of the 13th International Joint Conference on Artificial Intelligence*. Chambéry, France, August 28 - September 3, 1993. San Mateo, CA: Morgan Kaufmann, pp.1172-1178.
- [7] V. Basili, G. Caldiera, D. Rombach, Experience Factory. In: J.J. Marciniak (ed.) *Encyclopedia of Software Engineering*, Vol. 1, pp. 469-476, John Wiley & Sons, New York, 1994.
- [8] R. Benjamins, D. Fensel, S. Decker. KA2: Building Ontologies for the Internet: A Midterm Report. *International Journal of Human Computer Studies*, 51(3):687, 1999.
- [9] T. Berners-Lee. *Weaving the Web*. Harper, 1999.
- [10] P. Clark, J. Thompson, B. Porter. Knowledge Patterns. In *KR-2000: Proceedings of the Conference on Knowledge Representation and Reasoning*. Breckenridge, CO, USA, 12-15 April 2000. San Mateo, CA: Morgan Kaufmann, 2000, pp. 591-600.
- [11] CRISP. <http://www.crisp-dm.org/CRISPPWP-0800.pdf>. Abgerufen am 5. September 2001.
- [12] D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, M. Vilain. Mixed-Initiative Development of Language Processing Systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 1997, Association for Computational Linguistics, 31 March - 3 April, Washington D.C., U. S. A.
- [13] S. Decker, D. Brickley, J. Saarela, J. Angele. A query and inference service for RDF. In *QL'98 - The Query Languages Workshop*. W3C, 1998. <http://www.w3.org/TandS/QL/QL98/>. Abgerufen am 5. September 2001.

- [14] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, I. Horrocks. The Semantic Web: the roles of XML and RDF. *IEEE Internet Computing*, 4(5), Sept.-Oct. 2000, pp. 63 –73.
- [15] A. Dengel, A. Abecker, A. Bernardi, L. van Elst, H. Maus, S. Schwarz, M. Sintek. Konzepte zur Gestaltung von Unternehmensgedächtnissen. *Künstliche Intelligenz*, Themenheft Wissensmanagement, 1/2002. In Druck.
- [16] A. Doan, P. Domingos, A. Halevy. Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. *Proceedings of ACM SIGMOD 2001*. ACM Press, 2001.
- [17] D. Ellis, J. Furner-Hines, P. Willett, On the Measurement of Inter-Linker Consistency and Retrieval Effectiveness in Hypertext Databases. *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. Dublin, Ireland, 3-6 July 1994. ACM/Springer 1994, pp. 51-60.
- [18] M. Erdmann, A. Maedche, H.-P. Schnurr, S. Staab. From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools. In *ETAI Journal - Section on Semantic Web (Linköping Electronic Articles in Computer and Information Science)*, 6, 2001.
- [19] M. Erdmann, R. Studer. How to Structure and Access XML Documents with Ontologies. *Data and Knowledge Engineering* (Special issue on intelligent information integration), 36(3): 317-335, 2001.
- [20] D. Foskett. Thesaurus. In: *Readings in Information Retrieval*. San Francisco, CA: Morgan Kaufmann, 1997.
- [21] T. R. Gruber. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*. 6(2): 199-221, 1993.
- [22] U. Hahn and K. Schnattinger. Towards Text Knowledge Engineering. In: *Proceedings of AAAI '98*. Cambridge/Menlo Park: MIT Press/AAAI Press, 1998, pp. 129-134.
- [23] U. Hahn, S. Schulz, M. Romacker: Part-Whole Reasoning: A Case Study in Medical Ontology Engineering. *IEEE Intelligent Systems*, 14(5): 59-67, 1999.
- [24] S. Handschuh, S. Staab, A. Maedche. CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework. In: *K-CAP 2001 – Proceedings of the First International ACM Conference on Knowledge Capture*. October 21-23, 2001, Victoria, B.C., Canada.
- [25] S. Handschuh, S. Staab. Authoring and Annotation of Web Pages in CREAM. Submitted to WWW2002.

- [26] A. Hotho, A. Maedche, S. Staab, R. Studer. SEAL-II — The Soft Spot Between Richly Structured and Unstructured Knowledge. *Journal of Universal Computer Science*, 7(7): 566-590. Springer, 2001.
- [27] A. Hotho, A. Maedche, S. Staab. Ontology-based Text Document Clustering. To appear in: *Künstliche Intelligenz*, Themenheft Text Mining, 2&3/2002.
- [28] M. Jarke. Wissenskontexte. *Künstliche Intelligenz*, Themenheft Wissensmanagement, 1/2002. In Druck.
- [29] T. Joachims. The Maximum-Margin Approach to Learning Text Classifiers: Methods, Theory, and Algorithms. Dissertation. Fachbereich Informatik, Universität Dortmund, 2001.
- [30] R. Kaplan, D. Norton. The Balanced Scorecard: Learning and Growth Perspective. . In D. Morey et. al. (eds.) *Knowledge Management – Classic and Contemporary Works*. Cambridge, MA: MIT Press, 2001, pp. 139-183. Reprint from: *The Balanced Scorecard: Translating Strategy into Action*. Boston: Harvard Business School Press, 1996.
- [31] M. Kifer, G. Lausen, J. Wu. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843.
- [32] M. Klettke, M. Bietz, I. Bruder, A. Heuer, D. Priebe, G. Neumann, M. Becker, J. Bedersdorfer, H. Uszkoreit, A. Mädche, S. Staab, R. Studer. GETESS – Ontologien, objektrelationale Datenbanken und Textanalyse als Bausteine einer semantischen Suchmaschine. *Datenbank-Spektrum*, 1(1): 14-24, 2001.
- [33] F. Kurfess, M. Hilario (eds.). *Proceedings of the ICDM-01 workshop on Integrating Data Mining and Knowledge Management*. November 29, 2001, San Jose, California, USA.
- [34] L. Leonard. *Inter-Indexer Consistence Studies, 1954-1975: A Review of the Literature and Summary of the Study Results*. Graduate School of Library Science, University of Illinois. Occasional Papers No.131, 1977.
- [35] M. F. López, A. Gómez-Pérez, J. Sierra, A. Sierra. Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems*, 14(1): 37-46, Jan./Feb. 1999.
- [36] A. Mädche, G. Neumann, S. Staab. Bootstrapping an Ontology-based Information Extraction System for the Web. To appear in: P.S. Szczepaniak, J. Segovia, J. Kacprzyk, L.A. Zadeh (eds.). *Intelligent Exploration of the Web*. Series "Studies in Fuzziness and Soft Computing". Physica-Verlag, Heidelberg.
- [37] A. Mädche, S. Staab, R. Studer. Ontologien. In *Wirtschaftsinformatik (WI-Schlagwort)*, 43(4): 393-396, 2001.

- [38] A. Mädche, S. Staab. Discovering Conceptual Relations from Text. In: W.Horn (ed.). *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, Berlin, August 21-25, 2000. IOS Press, Amsterdam, 2000, pp. 321-324.
- [39] A. Mädche, S. Staab. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2): 72-79, March/April 2001 (Special issue on the semantic web).
- [40] A. Mädche, S. Staab. Applying Semantic Web Technologies for Tourism Information Systems. In: Karl Wöber, Andrew Frew, Martin Hitz (eds.), *Proceedings of the 9th International Conference for Information and Communication Technologies in Tourism, ENTER 2002*. Springer Verlag, Innsbruck, Austria, 23 - 25th January 2002.
- [41] H. Maurer, M. Sapper. E-Learning Has to be Seen as Part of General Knowledge Management, In *Proceedings of ED-MEDIA 2001 World Conference on Educational Multimedia, Hypermedia & Telecommunications*, Tampere, AACE, Charlottesville, VA, 2001, pp. 1249-1253.
- [42] N. Noy, W. Grosso, M. Musen. Knowledge-Acquisition Interfaces for Domain Experts: An Empirical Evaluation of Protégé-2000. *Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE2000)*, Chicago, IL, 2000.
- [43] C.K. Ogden, I.A. Richards. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. London: Routledge & Kegan Paul Ltd., 10th edition, 1923.
- [44] F. Pentaris, Y. Ioannidis. Interoperability via Mapping Objects. In: *Proceedings of the DELOS Workshop on Interoperability in Digital Libraries (at ECDL 2001)*. To appear as ERCIM. Workshop Report <http://www.ercim.org/>
- [45] G. Probst, K. Romhardt, S. Raub. *Wissen managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. Wiesbaden: Gabler, 1999.
- [46] A. Rabarijaona, R. Dieng, O. Corby, R. Ouaddari. Building and searching an XML-based corporate memory. *IEEE Intelligent Systems*, 15(3): 56-63, May-June 2000.
- [47] U. Reimer, A. Margelisch, M. Staudt. EULE. A Knowledge-Based System to Support Business Processes. *Knowledge Based Systems*, 13(5): 261-269, 2000.
- [48] P. Senge. Reflection on “A Leader’s New Work: Building Learning Organizations”. In D. Morey et. al. (eds.) *Knowledge Management – Classic and Contemporary Works*. Cambridge, MA: MIT Press, 2001, pp. 54-60.
- [49] K. von Schierstedt. *Wissensmanagement bei SD&M*. Folien eines Vortrags auf der WM-2001 - 1. Konferenz Professionelles Wissensmanagement – Erfahrungen und

- Visionen, Baden-Baden, März 2001. Vgl. [50]. Download von <http://wm2001.aifb.uni-karlsruhe.de/InvitedTalks/SchierstedtSD&M.ppt>. Abgerufen am 5. September 2001.
- [50] H.-P. Schnurr, S. Staab, R. Studer, G. Stumme, Y. Sure (eds.). *WM2001 – Professionelles Wissensmanagement – Erfahrungen und Visionen. Proceedings der Konferenz WM2001*, Shaker Verlag, März 2001.
 - [51] A. Schreiber, B. Dubbeldam, J. Wielemaker, B. Wielinga. Ontology-based Photo Annotation. *IEEE Intelligent Systems*, 16(3), 2001, pp. 66-74.
 - [52] A. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. Van de Velde, and B. Wielinga. *Knowledge Engineering and Management — the CommonKADS Methodology*. The MIT Press, Cambridge, Massachusetts; London, England, 1999.
 - [53] H. Schuetze, C. Silverstein. Projections for efficient document clustering. In *Proc. of SIGIR-1997*, Philadelphia, PA, July 1997, pages 74-81. Morgan Kaufmann, 1997.
 - [54] J. Sowa. *Knowledge Representation. Logical, Philosophical and Computational Foundations*. Pacific Grove, CA: Brooks Cole, 2000. (vgl. auch <http://www.jfsowa.com/ontology/>).
 - [55] J. Sowa (ed.). *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, San Mateo, CA: Morgan Kaufmann, 1991.
 - [56] S. Staab. *Grading Knowledge – Extracting Degree Information from Texts*. LNCS/LNAI 1744. Heidelberg, Berlin: Springer Verlag, December 1999.
 - [57] S. Staab. From Binary Temporal Relations to Non-Binary Ones and Back. *Artificial Intelligence*, 128(1-2): 1-29, Elsevier 2001.
 - [58] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Mädche, H.-P. Schnurr, R. Studer, Y. Sure. Semantic Community Web Portals. In: *WWW9 / Computer Networks (Special Issue: WWW9 - Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, May, 15-19, 2000)*, 33(1-6): 473-491. Elsevier, 2000.
 - [59] S. Staab, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, & B. Wrenger. GETESS - Searching the Web Exploiting German Texts. *CIA '99 - Proceedings of the 3rd international Workshop on Cooperating Information Agents*. Upsala, Sweden, July 31-August 2, 1999, pp.113-124, LNAI 1652, Berlin, Heidelberg: Springer.

- [60] S. Staab, M. Erdmann, A. Mädche. Ontologies in RDF(S). In *ETAI Journal - Section on Semantic Web (Linköping Electronic Articles in Computer and Information Science)*, 6, 2001.
- [61] S. Staab, M. Erdmann, A. Mädche. Engineering Ontologies Using Semantic Patterns. In A. Preece & D. O'Leary (eds.), *Proceedings of the IJCAI-01 Workshop on E-Business & the Intelligent Web*. Seattle, WA, USA, August 5, 2001.
- [62] S. Staab, A. Mädche, S. Handschuh. Creating Metadata for the Semantic Web: An Annotation Framework and the Human Factor. Institute AIFB, University of Karlsruhe. Technical report, 2000.
- [63] S. Staab, A. Mädche. Knowledge Portals – Ontologies at Work. *AI Magazine*, 22(2), Summer 2001, pp. 63-75.
- [64] S. Staab, H.-P. Schnurr. Smart Task Support through Proactive Access to Organizational Memory. *Knowledge-based Systems*, 3(5): 251-260. Elsevier, 2000.
- [65] S. Staab, H.-P. Schnurr, R. Studer, Y. Sure. Knowledge Processes and Ontologies. *IEEE Intelligent Systems*, 16(1): 26-35, January/February 2001 (Special issue on knowledge management).
- [66] S. Staab, R. Studer, Y. Sure. Knowledge Processes and Meta Processes in Ontology-based Knowledge Management. Eingereicht zu: C. Holsapple (ed.) *Handbook on Knowledge Management*. International Handbooks on Information Systems, Springer Verlag, 2002.
- [67] L. Stojanovic, S. Staab, R. Studer. eLearning based on the Semantic Web. *WebNet 2001 - World Conference on the WWW and the Internet*. Orlando, Florida, USA, Oct. 23-27, 2001.
- [68] N. Stojanovic, A. Maedche, S. Staab, R. Studer, Y. Sure. SEAL — A Framework for Developing SEmantic PortALs. In: *K-CAP 2001 – Proceedings of the First International ACM Conference on Knowledge Capture*. October 21-23, 2001, Victoria, B.C., Canada.
- [69] K. Sycara, M. Klusch, S. Widoff, J. Lu. Dynamic Service Matchmaking Among Agents in Open Information Environments. *SIGMOD Record*, 28(1): 47-53 (1999)
- [70] Y. Sure, A. Mädche, S. Staab. Leveraging Corporate Skill Knowledge - From ProPer to OntoProper. In: D. Mahling & U. Reimer. *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management*. Basel, Switzerland, October 30-31, 2000, pp. 22.1-22.9.

- [71] Y. Sure. A Tool supported Methodology for Ontology-based Knowledge Management. In *Proc. of the Annual Meeting of the FG Wissensmanagement (FGWM) 2001*, Dortmund, Germany, 10th-11th October 2001. <http://research.swisslife.ch/gi/fgwm2001/>.
- [72] Y. Sure, R. Studer. On-To-Knowledge Methodology – Evaluated and Employed Version. Deliverable 16. On-To-Knowledge Project Report, 2001.
- [73] H. Takeuchi, I. Nonaka. Theory of Organizational Knowledge Creation. In D. Morey et. Al. (eds.) *Knowledge Management – Classic and Contemporary Works*. Cambridge, MA: MIT Press, 2001, pp. 139-183. Reprint from: *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation*. New York: Oxford University Press, 1995, pp. 56-94.
- [74] C. Tautz, K-D. Althoff. Using Case-Based Reasoning for Reusing Software Knowledge. In: *Case-Based Reasoning Research and Development, Second International Conference, ICCBR-97*, Providence, Rhode Island, USA, July 25-27, 1997. LNCS 1266, Springer, 1997, pp. 156-165.
- [75] TopicMaps.Org XTM Authoring Group. *XTM: XML Topic Maps (XTM) 1.0: TopicMaps.Org Specification*, 3 Mar 2001. <http://www.topicmaps.org/xtm/1.0/>. Abgerufen am 1. September, 2001.

Knowledge Processes and Ontologies

Steffen Staab and Rudi Studer, *Institute AIFB—University of Karlsruhe and Ontoprise GmbH*
Hans-Peter Schnurr, *Ontoprise GmbH*
York Sure, *Institute AIFB—University of Karlsruhe*

Increases in product complexity, the move toward globalization, the emergence of virtual organizations, and the increase in focus on customer orientation all demand a more thorough and systematic approach to managing knowledge. Knowledge management is a major issue for human resource management, enterprise organization, and

enterprise culture. But IT plays a major supporting role in managing knowledge.

You typically build IT-supported KM solutions around an organizational structure that integrates informal, semiformal, and formal knowledge to facilitate its access, sharing, and reuse.¹ In such contexts, where knowledge has to be modeled, structured, and interlinked, ontologies can help formalize the knowledge shared by a group of people.²

In this article, we present an approach for ontology-based KM that includes a suite of ontology-based tools as well as a methodology for developing ontology-based KM systems. Our approach, shown in Figure 1, builds on the distinction between *knowledge process* (handling knowledge items) and *knowledge metaprocess* (introducing and maintaining KM systems).

Ontologies constitute the glue that binds knowledge subprocesses together. Ontologies open the way to move from a document-oriented view of KM to a content-oriented view, where knowledge items are interlinked, combined, and used. The method for developing KM systems that we outline in this article (that is, the knowledge metaprocess) extends and improves the CommonKADS method³ by introducing specific guidelines for developing and maintaining ontologies.

Our approach shows that you can clearly identify and handle different subprocesses that drive the development and use of KM applications. You support these subprocesses by appropriate tools that are tied together by the ontology infrastructure.⁴

Knowledge items versus documents

Companies have typically pursued a very simple, pragmatic approach for introducing KM into the enterprise (moving along the knowledge metaprocess in the yellow circle in Figure 1), which means that they have picked only the low-hanging fruit. We summarize this approach in the left column of Figure 2. What appears preeminent in this approach is the focus on handling documents (steps 2 and 3) and the existing but minor role of the appendix “process.”

In spite of its immediate successes, this approach presents several disadvantages. In particular, it often leads to the consequence that the knowledge process steps (the blue circle) of creation, import, captur-

This approach for ontology-based knowledge management includes a tool suite and methodology for developing ontology-based KM systems. It builds on the distinction between knowledge process and knowledge metaprocess and is illustrated by CHAR, a knowledge management system for corporate history analysis.

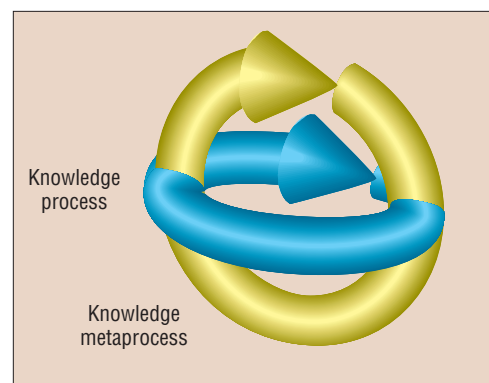


Figure 1. Two orthogonal processes with feedback loops illustrate our approach to knowledge management.

ing, retrieving and accessing, and using are only very loosely connected, if at all. The underlying reason is that for each of these steps different types of business documents play a major role, which makes “knowledge re-use”—let alone knowledge refinding—extremely difficult.

Relevant knowledge items can appear in a multitude of different document formats: text documents, spreadsheets, presentation slides, database entries, Web pages, construction drawings, or email, to name but a few. The challenge lies in how you handle the knowledge.

At the one extreme, in traditional document management, IT support for KM cannot take advantage of the document contents themselves but only their explicit or implicit classification. At the other extreme, there are expert systems that structure and codify all knowledge in the system. However, although such an approach might sometimes be appropriate, not everything can be codified. A lot of knowledge is created sporadically, and the value of its reuse can only be demonstrated over time. You must therefore search for an adequate balance between reuse, level of formality, and cost. For instance, certain help desk scenarios imply long-term use of extremely well-defined knowledge items.⁵ In these cases, it might be economically advantageous to spend some time and money on coding. On the other hand, a hallway discussion is usually not worth codifying at all, because it might not be reusable.

To balance conflicting needs and manage various degrees of encoded knowledge, it can be useful to employ the concept of metadata, which we can define in two complementary ways:

- *Data describing issues related to the content of data.* We divide this category into two orthogonal dimensions: the formality of the data and the containment of the metadata. In the first dimension, metadata might range from very informal descriptions of documents (like free-text summaries of books) to very formal descriptions (like ontology-based document annotation). In the second dimension, parts of metadata might be internal to the data that is described (like an HTML author tag) while others might be stored completely independently from the document they describe (like a bibliography database that classifies the documents it refers to but does not contain them).

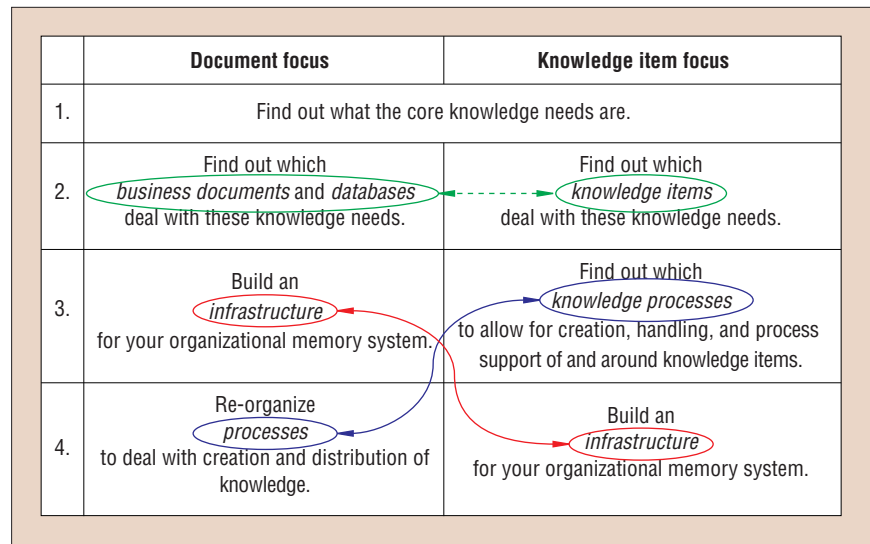


Figure 2. The document focus compared to the knowledge item focus.

- *Data that describes the structure of data.* In our case, you can call this type of metadata “meta metadata” because it describes the structure of metadata. This distinction boils down to an ontology that formally describes the domain of the KM application, possibly including parts of the organization and the information structures.⁶

Metadata can help condense and codify knowledge for reuse in other steps of the KM process. It can also help link knowledge items of various degrees of formality together, thus allowing a sliding balance between depth of coding and costs.

Knowledge process

Once you fully implement a KM system, knowledge processes essentially revolve around the following steps (see Figure 3):

- *Creation or import.* The contents need to be created or converted so that they fit the conventions of the company.
- *Capture.* Knowledge items have to be captured in order to determine their importance and how they mesh with the company’s vocabulary conventions.
- *Retrieval and access.* This step satisfies the searches and queries for knowledge by the knowledge worker.
- *Use.* The knowledge worker will not only

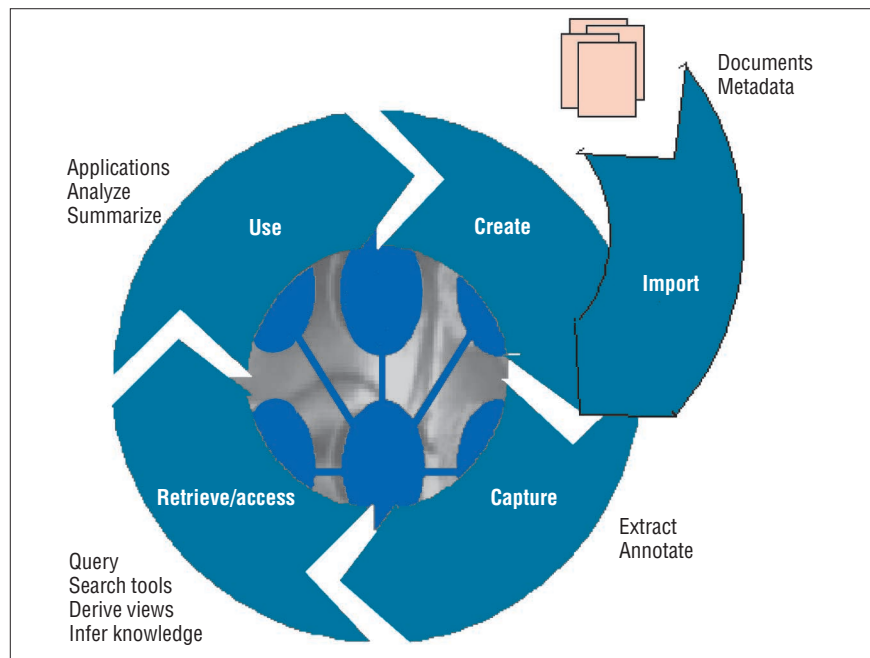


Figure 3. The knowledge process.

Table 1. Degrees of formal and informal knowledge.

Degree	Model	Interface	Example
Thoroughly formal	Relational	Form interface	Database interface
Formal	Content-structured document	Tight XML structure	XML-EDI
Partially formal	Document template	Loose XML structure	Investment recommendation template
Informal	Free text	No predefined structure	ASCII text file

recall knowledge items, but will process them for further use.

Knowledge creation

Creation of computer-accessible knowledge typically moves between the two extremes of formal and informal. Comparatively deep codification can often be done without requiring any extra effort. Business documents, in general, do not arbitrarily change but often come with some inherent structure, like engineering requirements for quality management. Our idea is to embed the structure of knowledge items into *document templates*, which are then filled on the fly by doing daily work.⁷

The granularity of this knowledge then lies in the middle between the extremes of coarsely representing business documents and representing them too finely. As Table 1 illustrates, there are several degrees of formality between formal and informal knowledge.

In Table 1, we use the term *content-structured documents* to refer to XML structures that are tightly (sometimes explicitly, sometimes implicitly) linked to a domain model. For instance, XML-EDI documents come with a predefined structure alluding to a standard framework for exchanging data, such as

invoices, healthcare claims, or project status reports. By the term *document templates*, we mean similar structures that come with a larger degree of freedom, including large chunks of informal knowledge items such as the one found in Figure 4.

Careful analysis of the in-use knowledge items lets you add formal knowledge parts into the process of creating documents. Doing so pushes the degree of formality slightly upwards without endangering overall system use.

Knowledge import

For many KM purposes, importing knowledge items into the KM system has the same or more importance than creating them. You can liken the situation to data warehousing, except that the input structures are more varied and the target structures are much richer and more complex.

For imported knowledge, accurate access to relevant items plays an even more important role than for homemade knowledge. For homemade knowledge items, people might act as a backup index, but they can't for recently imported knowledge that no one has yet seen. In fact, studies have shown that the parts that cover imported knowledge are less

heavily exploited than those covering home-grown ones.⁸

Knowledge capture

Once you create knowledge items, the next step is to capture their essential contents. There are several indexing and abstracting techniques typical of library science. In addition, we provide a means to capture document excerpts as well as interlinkage between excerpts by our tool, OntoAnnotate.⁴

OntoAnnotate, illustrated in Figure 5, lets you create objects and describe them with their attributes and relations. Describing objects this way helps exploit knowledge found on Web pages, in spreadsheets, or in text documents. In the example shown in Figure 5, OntoAnnotate captures several facts, such as company M.A. Hanna selling its Shape Distribution Business to GE.

Through this annotation process, you create metadata that conform to the ontology and that can be aligned with related information to yield analyses and derivations like those we describe in this article. The origins of the metadata may be used to validate the overall information.

Knowledge retrieval and access

You typically perform large parts of knowledge retrieval and access through a conventional GUI. You can use the ontology to derive further views of the knowledge. In particular, you can exploit the ontology for navigation purposes, offering navigation structures the way Yahoo does. Knowledge workers can explore what is in the organization's collective memory without being required to ask a particular question—which is often a hard task for newcomers.

Also, using an ontology lets you derive additional links and descriptions. For example, the ontology can help you derive state descriptions for points in time for which no explicit data exists (such as the current structure of a company that has changed its organization by mergers or acquisitions). Thus, it can provide new hyperlinks that aren't explicitly given. Ontologies can help complete views of your captured knowledge

<investmentrecommendation>		
<author>	Henrik Oppermann	</author>
<plandate>	October 18th, 2003	</plandate>
<interviewpartners>		
	<name> York Sure	</name>
	<name> Hans-Peter Schnurr	</name>
	<name> Steffen Staab	</name>
</interviewpartners>		
<recommend>	strong buy	</recommend>
<details>		
	<peergroup> ...	</peergroup>
	<...> ...	<...>
</details>		
</investmentrecommendation>		

Figure 4. An example of a knowledge item, in this case an investment recommendation template.

without requiring that all information is actually given.

Knowledge use

Knowledge use, which deals with the most intricate points of KM, is the part that is most often neglected. Many KM systems assume that once some relevant document is found, everything is done. Eventually, however, the way to use knowledge from the organization's collective memory becomes quite involved. Topics such as proactive access, personalization, and in particular, tight integration with subsequent applications play a crucial role for the effective reuse of knowledge. Very often it is not even the knowledge itself that is of most interest, but the derivations that can be made from the knowledge. For instance, in our case study, no single knowledge items about a company might be relevant to a market analyst, but the overall picture presented by the analysis can be quite relevant.

In addition, usage data tells a lot about the organization's memory and about the organization itself. You can analyze which processes, customers, and techniques are tied to core figures of the organization. In one of our cases, we identified the important role of a sales representative in South America only by his pervasive knowledge stored in the organizational memory about this market. Without the representative, the South American market might have gotten lost and, hence, the company began initiatives to avoid this situation.

Knowledge metaprocess

Over more than a decade, CommonKADS has been generally successful as a methodology for developing KM systems.³ It took until the 1990s for research to begin identifying the virtues of ontologies for sharing and reusing knowledge. But there is little work that tightly integrates ontology development into an overarching methodology for introducing knowledge management systems, such as CommonKADS. In contrast to seminal related work in the area of ontology development,^{2,9} our focus lies on the application-oriented development of ontologies, including feedback and requirements from the KM application.

Our model for ontology development ranges from the early stages of setting up a KM project to the final rollout of the ontology-based KM application. In this article, we integrate some lessons learned from our experiences into the steps to be taken to perform ontology development activities. Figure 6 shows the ontology development process.

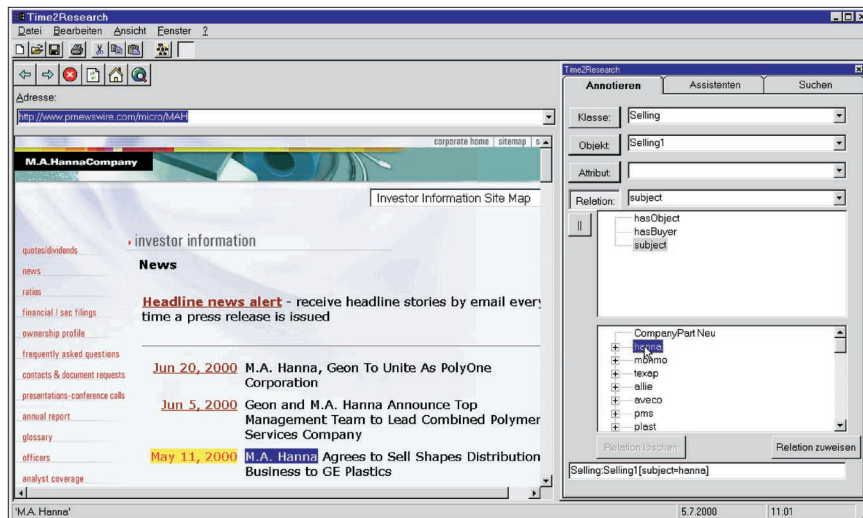


Figure 5. How OntoAnnotate works when capturing knowledge about a sale.

Feasibility study

Any KM system can only function satisfactorily if it is properly integrated. Several factors other than technology can determine success or failure. To analyze these factors, you have to perform a feasibility study to identify problem or opportunity areas and potential solutions. You also perform a feasibility study to put these problems or opportunities into a wider organizational perspective.

The feasibility study can help you determine economical and technical project feasibility. It can help you select the most promising focus area and the best solution to any potential problems. For our purposes, we adopted the kind of feasibility study described

in the CommonKADS methodology. The feasibility study should be carried out before actually developing ontologies because it serves as a basis for the kickoff phase.

Kickoff phase

The kickoff phase's output product is an ontology requirements specification document (see Figure 7). The kickoff phase describes what an ontology should support and sketches the planned area of the ontology application. It should also guide an ontology engineer to decide about inclusion, exclusion, and the hierarchical structure of concepts in the ontology. In this early stage, you should look for already developed and potentially reusable ontologies.

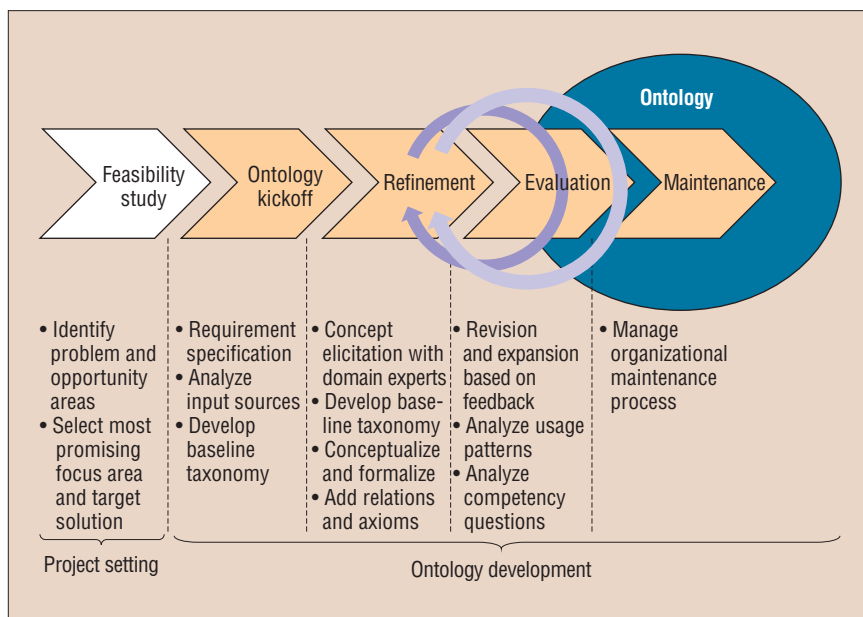


Figure 6. The ontology development process or knowledge metaprocess.

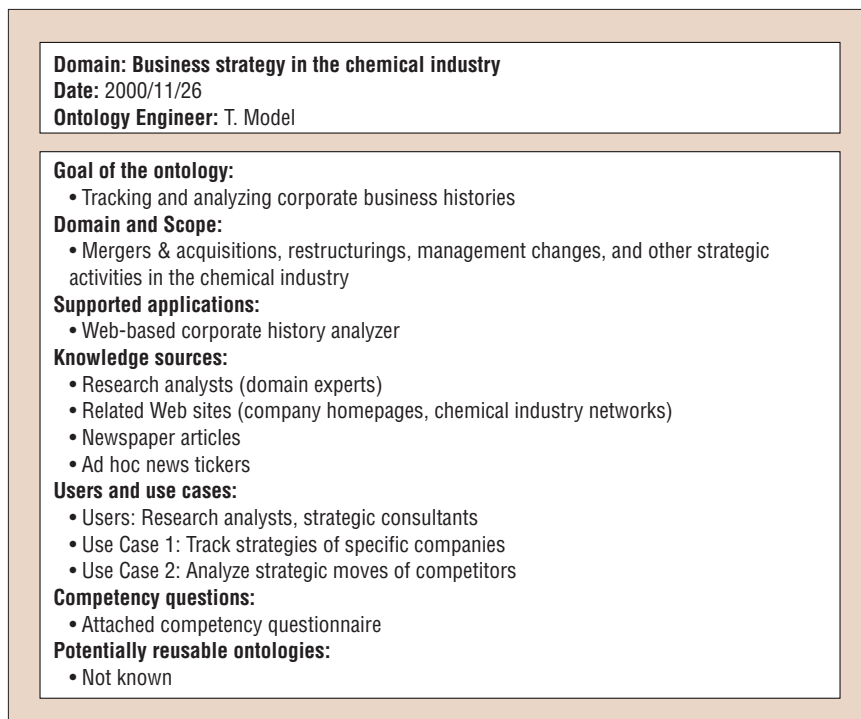


Figure 7. Ontology requirements specification document.

In summary, the kickoff phase should clearly detail several pieces of information:

- the ontology's goal,
- its domain and scope,
- the applications it supports,
- its knowledge sources (like domain experts, organization charts, business plans, dictionaries, index lists, or db-schemas), and
- its potential users and usage scenarios.

It should also include a competency questionnaire like the one shown in Table 2—essentially an overview of possible queries to the system that can indicate the scope and content of the domain ontology. Finally, the kickoff phase should detail any potentially reusable ontologies.

Refinement phase

The goal of the refinement phase is to produce a mature and application-oriented target ontology according to the specification given by the kickoff phase. We divide this phase into different subphases:

- Gathering an informal baseline taxonomy containing relevant concepts given during the kickoff phase.
- Eliciting knowledge from domain experts

based on the initial input from the baseline taxonomy to develop a seed ontology that contains relevant concepts and describes the relationships between them. The seed ontology is expressed at an epistemological level.

- Transferring the seed ontology into the *target ontology* expressed in formal representation languages like Frame Logic, Description Logic, or Conceptual Graphs.

Using potentially reusable ontologies (identified during the kickoff phase) can improve the speed and quality of the development during the whole refinement phase. These ontologies might provide useful hints for modeling decisions.

Evaluation phase

The evaluation phase serves as a proof of the usefulness of developed ontologies and their associated software environment. In this step, the ontology engineer checks whether the target ontology satisfies the ontology requirements specification document and whether the ontology supports or answers the competency questions analyzed in the kickoff phase of the project.

In this step you also test the ontology in the target application environment. Feedback from beta users might be valuable input for

further refinement of the ontology. The prototype system should track the ways users navigate or search for concepts and relations. You can then trace what areas of the ontology are used most often.

This phase is closely linked to the refinement phase. An ontology engineer might need to perform several cycles until the target ontology reaches the specified level. Rolling out the target ontology finishes the evaluation phase.

Maintenance phase

Specifications for ontologies often need to change to reflect changes in the real world. To reflect these changes, ontologies have to be maintained frequently. Maintaining ontologies is primarily an organizational process. There must be strict rules for the update-delete-insert processes within ontologies.

We recommend you gather changes to the ontology and initiate the switch-over to a new version of the ontology after thoroughly testing possible effects to the application. As in the refinement phase, feedback from users might be valuable for identifying changes.

Knowledge metaprocess instantiated

Actively tracking and managing relevant knowledge is a major task for knowledge-intensive companies. While the correct analysis of market situations and competitors are critical requirements for success, the failure to provide adequate knowledge about one's business environment can invite failure.

Feasibility study

In the feasibility study we found that management and professionals have a hard time gathering information, analyzing it, and performing their operational work. The task of the corporate researcher is to track relevant knowledge and communicate it to stakeholders within the company. Market analysts, consultants, and in-house market research departments try to track their industry's activities using traditional methods. Corporate research typically tracks newspaper articles, online databases, annual company reports, and competitors' Web pages and then presents the results to management.

There are several problems with the conventional research process:

- Information archives are document-based. For a collective gathering of facts, this is a document-centric view that is too coarse to be useful.

- Typical document management systems rely almost exclusively on information retrieval techniques that are inaccurate.
- Implications can only be made transparent if you use background knowledge, but systems today rarely support background knowledge.
- Different people might contribute knowledge.
- Different people might require different views of the same basic piece of information.

A KM system that covers such knowledge about the outside world should

- support the collective gathering of information on the level of facts rather than documents,
- integrate the gathering task smoothly into the common research process,
- allow you to intelligently combine facts,
- check new facts against the available background knowledge,
- allow multiple-view access to the knowledge through a single entry portal, and
- allow you to route derived facts back into the common workplace environment.

With these aims in mind, we developed an ontology-based application called the Corporate History Analyzer (CHAR).

Kickoff phase

The question was how to bring the required conceptual structures and reasoning capabilities into action following the further steps of our knowledge metaprocess shown in Figure 6. We collected user requirements in the requirements specification phase during structured interviews with corporate research analysts. The Ontology Requirements Specifications Document shown in Figure 7 is the product of this first phase.

We specified the ontology and then developed it in detail. We asked the domain experts what questions they would expect to be answered by a system supporting their corporate research work. These competency questions helped us find the most important concepts and relations between them.

Analyzing the questionnaires helped us conclude that the system should deliver answers about the acquisitions, mergers, and restructuring of companies over specific time periods. The questionnaire in Table 2 lists competency questions CQ1 to CQ4 that we compiled during the ontology kickoff. We also recognized the need for a clarification of organizational wording, so we could more clearly determine, for example, whether a business unit is a division or a department.

Refinement phase

In the refinement phase, we brought the

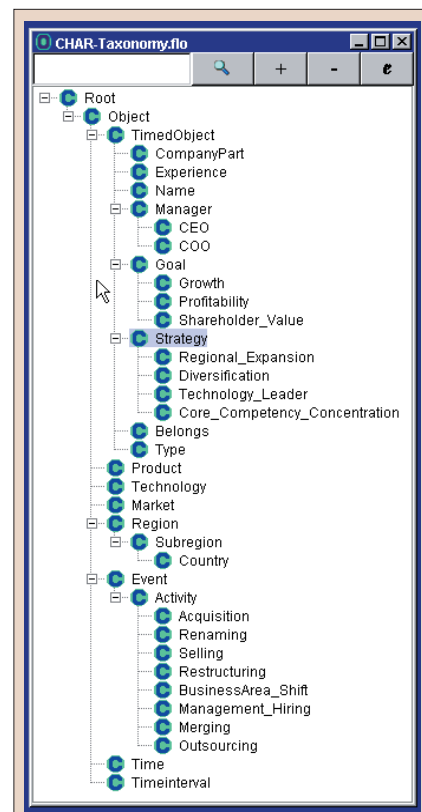


Figure 8. The Corporate History Analyzer taxonomy.

concepts into a taxonomic hierarchy designed for the knowledge engineer. We showed the taxonomy—a part of which is shown in Figure 8—to domain experts to generate additional concepts and relevant

Table 2. Sample competency questionnaire for a business strategy in the chemical industry.

Domain: Business strategy in the chemical industry

Date: 2000/11/26

Ontology Engineer: T. Model

CQ no.	Competency questions	Concepts	Relation
CQ1	What are the subsidiaries, divisions and locations of company X?	company, subsidiary, division, location	company <i>has</i> subsidiary company <i>has</i> division company <i>has</i> location
CQ2	Which companies acquired company X?	company, acquisition	company <i>makes</i> acquisition acquisition <i>has</i> buyer acquisition <i>has</i> seller
CQ3	Which companies merged in 1990 in the rubber industry?	company, merger, year, industry	company <i>makes</i> merger company <i>isPartOf</i> industry merger <i>happensIn</i> year
CQ4	Who is CEO of company X?	CEO, company	company <i>has</i> CEO
CQ5	Which activity of company X leads to operation in region Y?	activity, company, operation, region	company <i>performs</i> activity activity <i>leadsTo</i> operation operation <i>takesPlaceIn</i> region
CQ6	Is there any regional expansion of company X due to the acquisition of company Y?	expansion, company, region, acquisition	company <i>makes</i> expansion company <i>makes</i> acquisition expansion <i>takesPlaceIn</i> region
CQ7	...		

Figure 9. The Corporate History Analyzer.

attributes, and to establish relationships between the concepts. OntoEdit, our ontology engineering tool, supported this work.¹⁰ OntoEdit allows you to model the ontology and formalize it in different representation languages, like F-Logic and DAML+OIL.

We found that in the corporate history domain, time had to be modeled. A company has a beginning time and (potentially) an end time, which would be when the company gets acquired by another company, merges with another company, or goes bankrupt. So we found we had to model duration.

One of the key characteristics of CHAR is that the user provides facts to the system about actions like acquisitions and mergers and the system then infers the consequences. To do this, you have to model rules for all possible activities that incur such consequences. For instance, a rule might read like this: "If two companies are merged, a new company with a new name is created, the old companies cease to exist, they are from now on subsidiaries of the new company" or "If a division is outsourced it becomes part of a new company or it becomes a company on its own."

For CHAR, we had to develop a Web interface that used the ontology for querying and augmenting the knowledge repository (see Figure 9). We had to perform a query development step to formalize the views and competency questions described earlier. This

development step depended mostly on the actual application setting. It was independent from the ontology construction.

Evaluation phase

In the evaluation and testing phase, we investigated the ontology's usability. With regard to the CHAR ontology, we found that the competency questions CQ1 to CQ4 (Table 2) could be handled successfully by the ontology. However, we wanted complex knowledge content to be queried and depicted on one screen. This kind of content would include company purchases, regions of previous activities, and regions of activities of the purchased company.

The research analyst found that singleton knowledge items were spread over different screens of the application. They were very hard to combine by the research analyst to answer strategic questions. He wanted better support from the ontology. Through the discussion, we came up with additional competency questions (CQ5 and following) that should be handled by the ontology. We then added corresponding axioms to the ontology and introduced a new query window for strategic questions.

Maintenance phase

We are currently in a maintenance phase, where through a shift in the goal orientation of

the application, we face requirements for considerable extension of the ontology and the application. In addition to corporate histories, the system must now support the extended tracking of market circumstances, including a more detailed view of available technology and better comparisons with peer groups.

Major problems that we now must face include the documentation and versioning problems that have so far been neglected by practically all ontology development tools. We are working toward resolving these issues in our ontology environment OntoEdit.

We developed our methodology parallel to CHAR. Thus, we could not follow all the methodological steps we've presented here from the very beginning. Developers who are new to the project need to reverse-engineer some old ontology parts to find answers about how these parts actually work. We expect that comprehensive use of the proposed methodology and its full-fledged support by OntoEdit will mostly prevent these problems in the future.

Knowledge process instantiated

CHAR should allow many people to contribute factual knowledge in a way that can be embedded into their common work process and organized around a semantic background. It should also provide multiple views onto the same knowledge base for different time frames, for different regional foci, for varying intra-organizational structures, and for different strategic questions, to name but a few.

Providing knowledge

Providing new facts for the knowledge repository should be as easy and as smoothly integrated into common work as possible. There are several ways to do this:

- You can enter information through a form-based interface.
- When information is produced during documentation or report-writing, you can use a template approach to generate knowledge. Both of these tools affect the knowledge creation step.
- You can use wrapper mechanisms to provide data from tables and lists on the Web (import step).
- Most important for CHAR, you can use our annotation tool to add metadata to data given in documents, which affects the capturing step.

Figure 5 shows a snapshot of OntoAnno-

tate.⁴ The user works with documents using a text or spreadsheet tool or an Internet browser. When the user detects some relevant change being described in the document, he or she highlights the word or phrase and uses the annotation tool to select the type of the phrase (like “M.A.Hanna is a company”) and its relation to other material (like “Hanna sells Shapes Distribution Business to GE Plastics on 11 May 2000”). The knowledge repository stores the document, these facts, the time of annotation, and meta-data about the annotator.

Querying for knowledge

We designed CHAR’s query interface to deal with organizational and strategic questions that depend on spatiotemporal con-

straints. CHAR renders views that can be seen on a common Web browser. Actually, these views look just like common Web sites. The available menu selections are controlled by the knowledge of the back-end system. For example, you can select companies known to exist in the knowledge repository. Figure 9 shows the main views offered by CHAR. These views include Activities, Organization, Know-How, Strategic Questions, and General Query Possibilities (indicated by “Search”).

The first major category of queries relevant to the corporate history is about organizational structures and the activities that change organizational structures. For instance, the view of “Acquisitions of M.A.Hanna” returns all its purchases. CHAR offers correspond-

ing views for Sales, Mergers, Restructuring, and Management Changes (see Figure 10).

What is interesting to note at this point is that it is rather difficult to get a clear picture of what is really happening with M.A.Hanna. It is difficult and time-consuming for the human analyst to detect some trend in lists of single knowledge items. Observations become much easier when different types of knowledge items can be related and contrasted.

For instance, Figure 10 shows two snapshots of the organization of M.A.Hanna. They are automatically derived from single activities, like acquisitions and restructurings, and they can give the analyst a neat picture of how formerly isolated purchases that M.A.Hanna made before 1994 were more tightly integrated in the company in 1997 (for example, “Southwest Chemical” having been reorganized into the Business Area “Plastic Compounding”).

In addition to sophisticated support based on concrete facts and figures, CHAR supports strategic questions that indicate possi-

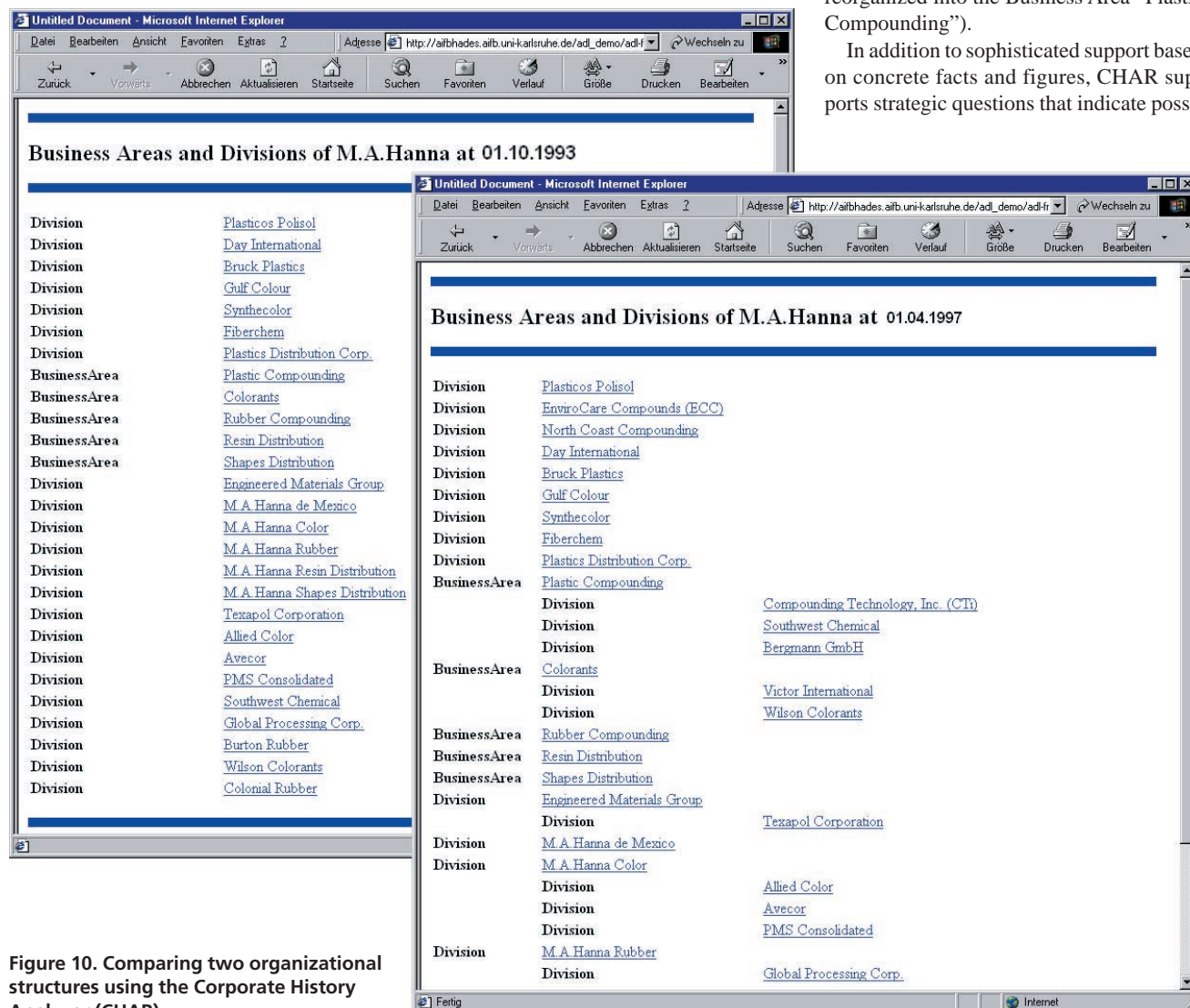


Figure 10. Comparing two organizational structures using the Corporate History Analyzer (CHAR).

ble answers to questions about business competitors that cannot be answered definitely and often rely on conjecture. For instance, the purchase of a company from abroad might lead to a gain of market share in that area, and thus to a regional diversification.

In the future, more and more companies will find that analysis of knowledge processes will feed back into the knowledge metaprocess cycle and can help to improve both. These companies will find that new concepts that come up during the use of a KM solution can be introduced back into the ontology to help it evolve. For this purpose, we investigate the use of semi-automatic ontology learning techniques.¹⁰

The challenges lie in analyzing KM processes and dynamically updating the KM solution. Our framework can help leverage these evolving systems by providing a concise view of the problem.

Related to this work, we are jointly organizing the first German conference on knowledge management, the WM2001 (<http://wm2001.aifb.uni-karlsruhe.de>). Steffen Staab and Rudi Studer are the coauthors, York Sure is the industrial liaison, and Hans-Peter Schnurr is the finance chair. ■

Acknowledgments

We thank our colleagues and students at AIFB, University of Karlsruhe, VU Amsterdam, and Ontoprise GmbH, including H. Akkermans, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, M. Maier-Collin, D. Oberle, and D. Wenke, without whom the research would not have been possible. The research presented in this paper was partially funded by EU in the IST-1999-10132 project On-To-Knowledge and by BMBF in the project GETESS (01IN901C0).

References

1. R. Dieng et al., "Methods and Tools for Corporate Knowledge Management," *Int'l J. Human-Computer Studies*, vol. 51, no. 3, 1999, pp. 567-598.
2. M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications," *The Knowledge Eng. Rev.*, vol. 11, no. 2, 1996, pp. 93-136.
3. G. Schreiber et al., *Knowledge Engineering and Management: The CommonKADS Methodology*, MIT Press, Cambridge, Mass., 1999.
4. S. Staab et al., "Semantic Community Web Portals," *Proc. WWW-9/Computer Networks*, Elsevier, New York, vol. 33, 2000, pp. 473-491.
5. L. Morgenstern, "Inheritance Comes of Age: Applying Nonmonotonic Techniques to Problems in Industry," *Artificial Intelligence*, no. 103, 1998, pp. 1-34.
6. A. Abecker et al., "Toward a Technology for Organizational Memories," *IEEE Intelligent Systems*, vol. 13, no. 3, May/June 1998, pp. 40-48.
7. S. Staab and D. O'Leary, eds., *Bringing Knowledge to Business Processes*, 2000 AAAI Spring Symposium technical report SS-00-03, AAAI Press, Menlo Park, Calif., 2000.
8. D. O'Leary, "Knowledge Management: An Analysis Based on Knowledge Use and Reuse," *IEEE Intelligent Systems*, vol. 16, no. 1, Jan./Feb., 2001.
9. M. F. López et al., "Building a Chemical Ontology Using Methontology and the Ontology Design Environment," *IEEE Intelligent Systems*, vol. 14, no. 1, Jan./Feb. 1999, pp. 37-46.
10. A. Maedche and S. Staab, "Mining Ontologies from Text," *Proc. 12th Int'l Conf. Knowledge Eng. and Knowledge Management, Lecture Notes in Computer Science*, vol. 1937, 2000, Springer Verlag, New York, pp. 189-202.

The Authors

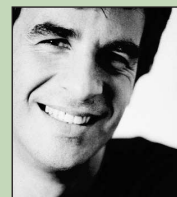


Steffen Staab is an assistant professor at the University of Karlsruhe and cofounder of Ontoprise GmbH. He received an MSE in computer science from the University of Pennsylvania and a Dr. rer.

nat. from the University of Freiburg, Computational Linguistics Lab. He has chaired several international workshops, including the AAAI 2000 spring symposium on Business Processes and KM. Contact him at Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany; sst@aifb.uni-karlsruhe.de.



Rudi Studer is a professor in applied computer science at the University of Karlsruhe. His research interests include knowledge management, intelligent Web applications, knowledge engineering, and knowledge discovery. He received a doctorate in mathematics and computer science and a Habilitation in computer science at the University of Stuttgart. He is a member of IEEE, ACM, AAAI, and GI. Contact him at University of Karlsruhe, Institute AIFB, D-76128 Karlsruhe, Germany; studer@aifb.uni-karlsruhe.de.



Hans-Peter Schnurr is managing director of Ontoprise GmbH, a company providing a wide range of semantic Web technologies. His current research interests include semantic Web and ontology-based applications. He received a diploma in industrial engineering and business economics from the University of Karlsruhe. He is cofounder of the German association for knowledge management (GfWM) and member of the German Association for Computer Science (GI). Contact him at Ontoprise GmbH, Haid-und-Neu Str. 7, 76131 Karlsruhe, Germany; +49 (0) 721 6635933; Fax +49 (0) 721 6635934; schnurr@ontoprise.de.



York Sure is a PhD candidate at the Institute of Applied Computer Science and Formal Description Methods at the University of Karlsruhe. His current research interests include knowledge management, semantic Web, and ontology-based applications. He received a diploma in industrial engineering from the University of Karlsruhe. He is a member of the German association for knowledge management (GfWM) and the German Association for Computer Science (GI). Contact him at Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany; sure@aifb.uni-karlsruhe.de.

Ontology Learning for the Semantic Web

Alexander Maedche and Steffen Staab, *University of Karlsruhe*

The Semantic Web relies heavily on formal ontologies to structure data for comprehensive and transportable machine understanding. Thus, the proliferation of ontologies factors largely in the Semantic Web's success. *Ontology learning* greatly helps ontology engineers construct ontologies. The vision of ontology learning that we propose

includes a number of complementary disciplines that feed on different types of unstructured, semistructured, and fully structured data to support semiautomatic, cooperative ontology engineering. Our ontology-learning framework proceeds through ontology import, extraction, pruning, refinement, and evaluation, giving the ontology engineer coordinated tools for ontology modeling. Besides the general framework and architecture, this article discusses techniques in the ontology-learning cycle that we implemented in our ontology-learning environment, such as ontology learning from free text, dictionaries, and legacy ontologies. We also refer to other techniques for future implementation, such as reverse engineering of ontologies from database schemata or learning from XML documents.

Ontologies for the Semantic Web

The conceptual structures that define an underlying ontology provide the key to machine-processable data on the Semantic Web. *Ontologies* serve as metadata schemas, providing a controlled vocabulary of concepts, each with explicitly defined and machine-processable semantics. By defining shared and common domain theories, ontologies help people and machines to communicate concisely—supporting semantics exchange, not just syntax. Hence, the Semantic Web's success and proliferation depends on quickly and cheaply constructing domain-specific ontologies.

Although ontology-engineering tools have matured over the last decade,¹ manual ontology acquisition remains a tedious, cumbersome task that can easily result in a knowledge acquisition bottleneck. When

developing our ontology-engineering workbench, *OntoEdit*, we particularly faced this question as we were asked questions that dealt with time (“Can you develop an ontology quickly?”), difficulty (“Is it difficult to build an ontology?”), and confidence (“How do you know that you’ve got the ontology right?”).

These problems resemble those that knowledge engineers have dealt with over the last two decades as they worked on knowledge acquisition methodologies or workbenches for defining knowledge bases. The integration of knowledge acquisition with machine-learning techniques proved extremely beneficial for knowledge acquisition.² The drawback to such approaches,³ however, was their rather strong focus on structured knowledge or databases, from which they induced their rules.

Conversely, in the Web environment we encounter when building Web ontologies, structured knowledge bases or databases are the exception rather than the norm. Hence, intelligent support tools for an ontology engineer take on a different meaning than the integration architectures for more conventional knowledge acquisition.⁴

In ontology learning, we aim to integrate numerous disciplines to facilitate ontology construction, particularly machine learning. Because fully automatic machine knowledge acquisition remains in the distant future, we consider ontology learning as semiautomatic with human intervention, adopting the paradigm of balanced cooperative modeling for constructing ontologies for the Semantic Web.⁵ With this objective in mind, we built an architecture that combines knowledge acquisition with machine learning, drawing on

The authors present an ontology-learning framework that extends typical ontology engineering environments by using semiautomatic ontology-construction tools. The framework encompasses ontology import, extraction, pruning, refinement, and evaluation.

resources that we find on the syntactic Web—free text, semistructured text, schema definitions (such as document type definitions [DTDs]), and so on. Thereby, our framework's modules serve different steps in the engineering cycle (see Figure 1):

- Merging existing structures or defining mapping rules between these structures allows *importing* and *reusing* existing ontologies. (For instance, Cyc's ontological structures have been used to construct a domain-specific ontology.⁶)
- Ontology *extraction* models major parts of the target ontology, with learning support fed from Web documents.
- The target ontology's rough outline, which results from import, reuse, and extraction, is *pruned* to better fit the ontology to its primary purpose.
- Ontology *refinement* profits from the pruned ontology but completes the ontology at a fine granularity (in contrast to extraction).
- The target application serves as a measure for validating the resulting ontology.⁷

Finally, the ontology engineer can begin this cycle again—for example, to include new domains in the constructed ontology or to maintain and update its scope.

Architecture

Given the task of constructing and maintaining an ontology for a Semantic Web application such as an ontology-based knowledge portal,⁸ we produced support for the ontology engineer embedded in a comprehensive architecture (see Figure 2). The ontology engineer only interacts via the graphical interfaces, which comprise two of the four components: the OntoEdit Ontology Engineering Workbench and the Management Component. Resource Processing and the Algorithm Library are the architecture's remaining components.

The OntoEdit Ontology Engineering Workbench offers sophisticated graphical means for manual modeling and refining of the final ontology. The interface gives the user different views, targeting the epistemological level rather than a particular representation language. However, the user can export the ontological structures to standard Semantic Web representation languages such as OIL (ontology interchange language) and DAML-ONT (DAML ontology language), as well as our own F-Logic-based extensions of RDF(S)—we use RDF(S) to refer to the combined technologies

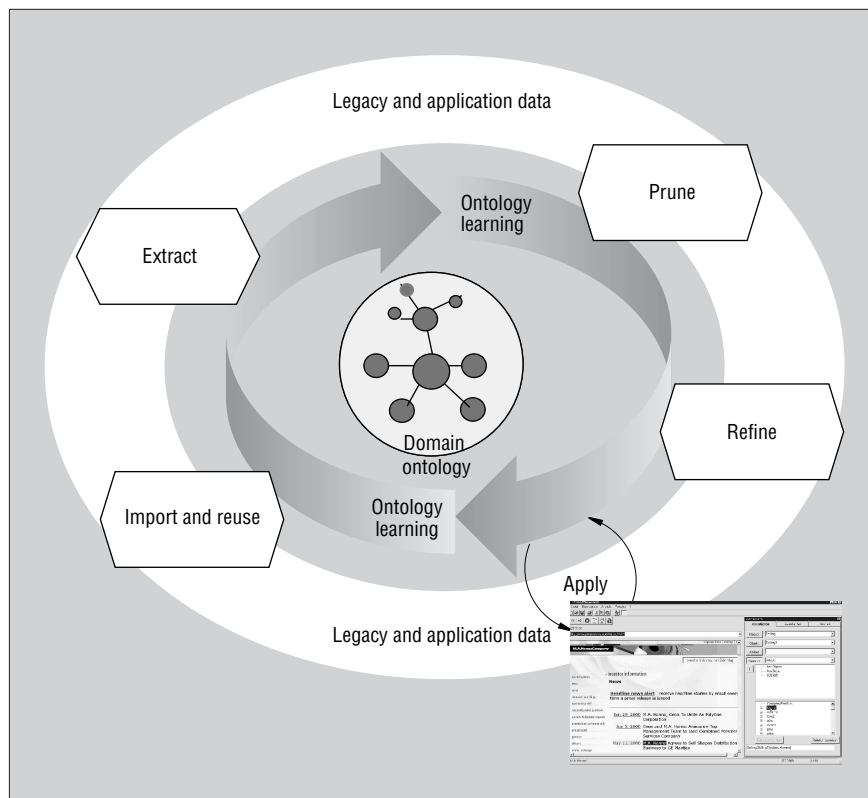


Figure 1. The ontology-learning process.

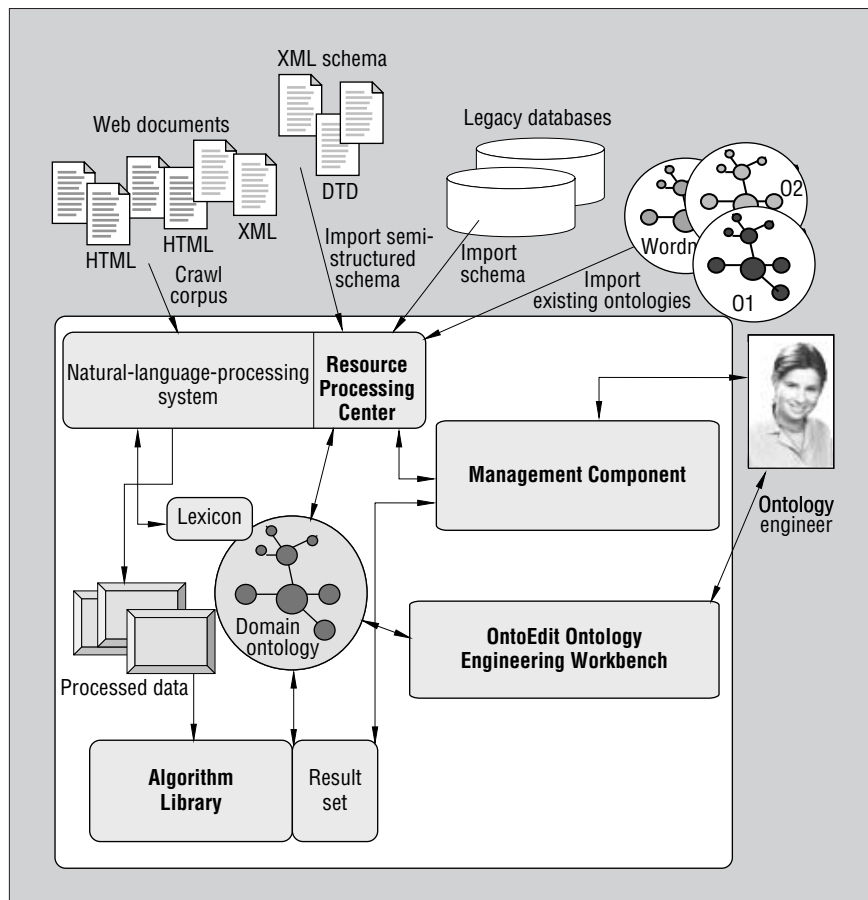


Figure 2. Ontology-learning architecture for the Semantic Web.

of the resource description framework and RDF Schema. Additionally, users can generate and access executable representations for constraint checking and application debugging through SilRi (*simple logic-based RDF interpreter*, www.ontoprise.de), our F-Logic inference engine, which connects directly to OntoEdit.

We knew that sophisticated ontology-engineering tools—for example, the Protégé modeling environment for knowledge-based systems¹—would offer capabilities roughly comparable to OntoEdit. However, in trying to construct a knowledge portal, we found that a large conceptual gap existed between the ontology-engineering tool and the input (often legacy data), such as Web documents, Web document schemata, databases on the Web, and Web ontologies, which ultimately determine the target ontology. Into this void we have positioned new components of our ontology-learning architecture (see Figure 2). The new components support the ontology engineer in importing existing ontology primitives, extracting new ones, pruning given ones, or refining with additional ontology primitives. In our case, the ontology primitives comprise

- a set of strings that describe lexical entries L for concepts and relations;
- a set of concepts C (roughly akin to synsets in WordNet⁹);
- a taxonomy of concepts with multiple inheritance (heterarchy) H_C ;
- a set of nontaxonomic relations R described by their domain and range restrictions;
- a heterarchy of relations— H_R ;
- relations F and G that relate concepts and relations with their lexical entries; and
- a set of axioms A that describe additional constraints on the ontology and make implicit facts explicit.⁸

This structure corresponds closely to RDF(S), except for the explicit consideration of lexical entries. Separating concept reference from concept denotation permits very domain-specific ontologies without incurring an instantaneous conflict when **merging** ontologies—a standard Semantic Web request. For instance, the lexical entry *school* in one ontology might refer to a building in ontology A, an organization in ontology B, or both in ontology C. Also, in ontology A, we can refer to the concept referred to in English by *school* and *school building* by the

German *Schule* and *Schulgebäude*.

Ontology learning relies on an ontology structured along these lines and on input data as described earlier to propose new knowledge about reasonably interesting concepts, relations, and lexical entries or about links between these entities—proposing some for addition, deletion, or merging. The graphical result set presents the ontology-learning process's results to the ontology engineer (we'll discuss this further in the "Association rules" section). The ontology engineer can then browse the results and decide to follow, delete, or modify the proposals, as the task requires.

Components

By integrating the previously discussed con-

In trying to construct a knowledge portal, we found that a large conceptual gap existed between the ontology-engineering tool and the input (often legacy data).

siderations into a coherent generic architecture for extracting and maintaining ontologies from Web data, we have identified several core components (including the graphical user interface discussed earlier).

Management component graphical user interface

The ontology engineer uses the management component to select input data—that is, relevant resources such as HTML and XML documents, DTDs, databases, or existing ontologies that the discovery process can further exploit. Then, using the management component, the engineer chooses from a set of resource-processing methods available in the resource-processing component and from a set of algorithms available in the algorithm library.

The management component also supports the engineer in discovering task-relevant legacy data—for example, an ontology-based crawler gathers HTML documents that are relevant to a given core ontology.

Resource processing

Depending on the available input data, the engineer can choose various strategies for resource processing:

- Index and reduce HTML documents to free text.
- Transform semistructured documents, such as dictionaries, into a predefined relational structure.
- Handle semistructured and structured schema data (such as DTDs, structured database schemata, and existing ontologies) by following different strategies for import, as described later in this article.
- Process free natural text. Our system accesses the natural-language-processing system Saarbrücken Message Extraction System, a shallow-text processor for German.¹⁰ SMES comprises a *tokenizer* based on regular expressions, a *lexical analysis* component including various word *lexicons*, an *amorphological analysis* module, a *named-entity recognizer*, a *part-of-speech tagger*, and a *chunk parser*.

After first preprocessing data according to one of these or similar strategies, the resource-processing module transforms the data into an algorithm-specific relational representation.

Algorithm library

We can describe an ontology by a number of sets of concepts, relations, lexical entries, and links between these entities. We can acquire an existing ontology definition (including L , C , H_C , R , H_R , A , F , and G), using various algorithms that work on this definition and the preprocessed input data. Although specific algorithms can vary greatly from one type of input to the next, a considerable overlap exists for underlying learning approaches such as association rules, formal concept analysis, or clustering. Hence, we can reuse algorithms from the library for acquiring different parts of the ontology definition.

In our implementation, we generally use a multistrategy learning and result combination approach. Thus, each algorithm plugged into the library generates normalized results that adhere to the ontology structures we've discussed and that we can apply toward a coherent ontology definition.

Import and reuse

Given our experiences in medicine,

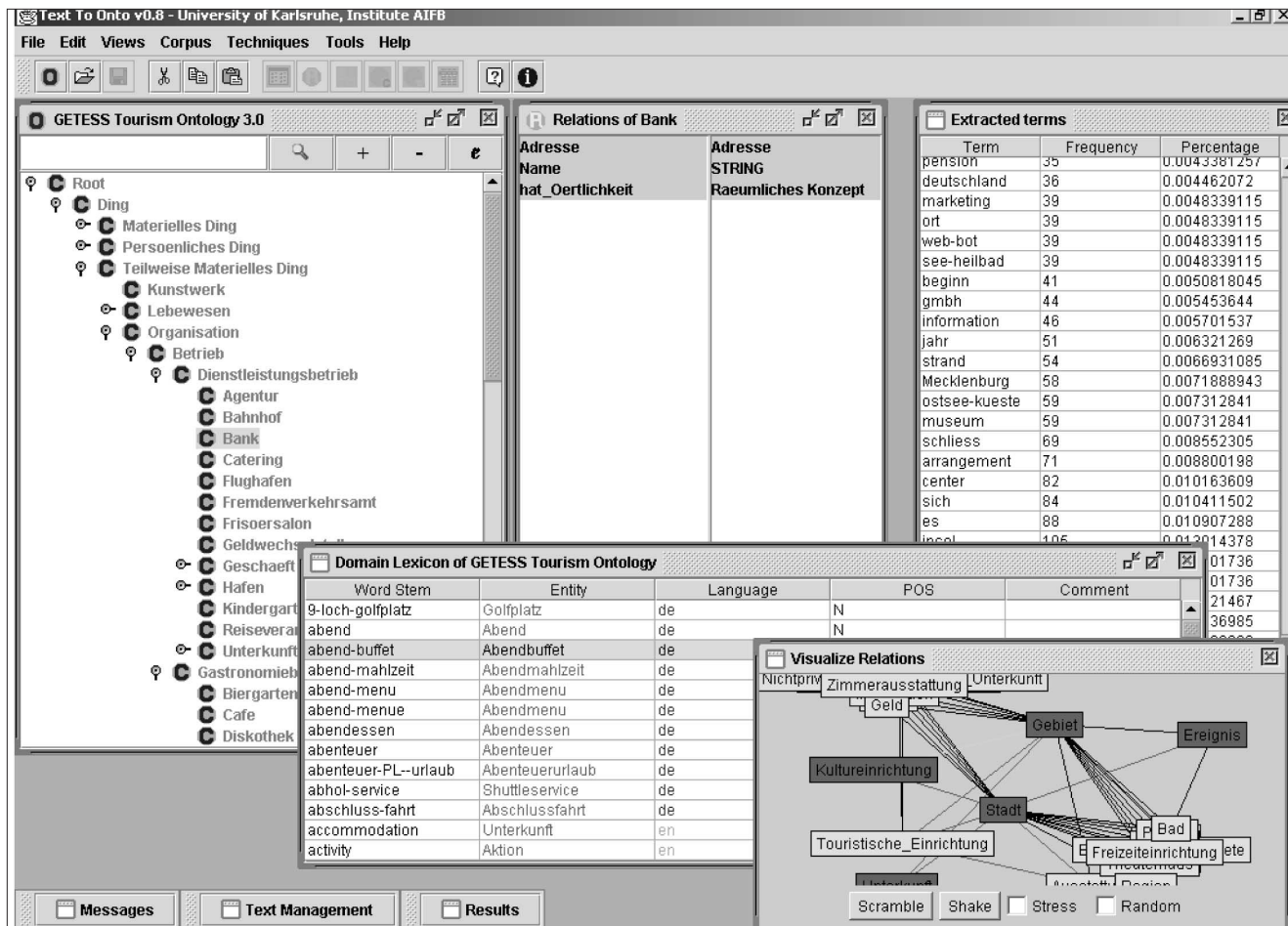


Figure 3. Screenshot of our ontology-learning workbench, Text-To-Onto.

telecommunications, tourism, and insurance, we expect that domain conceptualizations are available for almost any commercially significant domain. Thus, we need mechanisms and strategies to import and reuse domain conceptualizations from existing (schema) structures. We can recover the conceptualizations, for example, from legacy database schemata, DTDs, or from existing ontologies that conceptualize some relevant part of the target ontology.

In the first part of import and reuse, we identify the schema structures and discuss their general content with domain experts. We must import each of these knowledge sources separately. We can also import manually—which can include a manual definition of transformation rules. Alternatively, reverse-engineering tools—such as those that exist for recovering extended entity-relationship diagrams from a given database's SQL description (see the sidebar)—might facilitate the recovery of conceptual structures.

In the second part of the import and reuse step, we must merge or align imported con-

ceptual structures to form a single common ground from which to springboard into the subsequent ontology-learning phases of extracting, pruning, and refining. Although the general research issue of merging and aligning is still an open problem, recent proposals have shown how to improve the manual merging and aligning process. Existing methods mostly rely on matching heuristics for proposing the merger of concepts and similar knowledge base operations. Our research also integrates mechanisms that use an application-data-oriented, bottom-up approach.¹¹ For instance, formal concept analysis lets us discover patterns between application data and the use of concepts, on one hand, and their heterarchies' relations and semantics, on the other, in a formally concise way (see B. Ganter and R. Wille's work on formal concept analysis in the sidebar).

Overall, the ontology-learning import and reuse step seems to be the hardest to generalize. The task vaguely resembles the general problems encountered in data-warehousing—adding, however, challenging problems of its own.

Extraction

Ontology-extraction models major parts—the complete ontology or large chunks representing a new ontology sub-domain—with learning support exploiting various types of Web sources. Ontology-learning techniques partially rely on given ontology parts. Thus, we here encounter an iterative model where previous revisions through the ontology-learning cycle can propel subsequent ones, and more sophisticated algorithms can work on structures that previous, more straightforward algorithms have proposed.

To describe this phase, let's look at some of the techniques and algorithms that we embedded in our framework and implemented in our ontology-learning environment Text-To-Onto (see Figure 3). We cover a substantial part of the overall ontology-learning task in the extraction phase. Text-To-Onto proposes many different ontology learning algorithms for primitives, which we described previously (that is, *L*, *C*, *R*, and so on), to the ontology engineer building on several types of input.

Until recently, ontology learning—for comprehensive ontology construction—did not exist. However, much work in numerous disciplines—computational linguistics, information retrieval, machine learning, databases, and software engineering—has researched and practiced techniques that we can use in ontology learning. Hence, we can find techniques and methods relevant for ontology learning referred to as

- “acquisition of selectional restrictions,”^{1,2}
- “word sense disambiguation and learning of word senses,”³
- “computation of concept lattices from formal contexts,”⁴ and
- “reverse engineering in software engineering.”⁵

Ontology learning puts many research activities—which focus on different input types but share a common domain conceptualization—into one perspective. The activities in Table A span a variety of communities, with references from 20 completely different events and journals.

References

1. P. Resnik, *Selection and Information: A Class-Based Approach to Lexical Relationships*, PhD thesis, Dept. of Computer Science, Univ. of Pennsylvania, Philadelphia, 1993.
2. R. Basili, M.T. Pazienza, and P. Velardi, “Acquisition of Selectional Patterns in a Sublanguage,” *Machine Translation*, vol. 8, no. 1, 1993, pp. 175–201.
3. P. Wiemer-Hastings, A. Graesser, and K. Wiemer-Hastings, “Inferring the Meaning of Verbs from Context,” *Proc. 20th Ann. Conf. Cognitive Science Society (CogSci-98)*, Lawrence Erlbaum, New York, 1998.
4. B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag, Berlin, 1999.
5. H.A. Mueller et al., “Reverse Engineering: A Roadmap,” *Proc. Int’l Conf. Software Eng. (ICSE-00)*, ACM Press, New York, 2000, pp. 47–60.
6. P. Buitelaar, *CORELEX Systematic Polysemy and Underspecification*, PhD thesis, Dept. of Computer Science, Brandeis Univ., Waltham, Mass., 1998.
7. H. Assadi, “Construction of a Regional Ontology from Text and Its Use within a Documentary System,” *Proc. Int’l Conf. Formal Ontology and Information Systems (FOIS-98)*, IOS Press, Amsterdam.
8. D. Faure and C. Nedellec, “A Corpus-Based Conceptual Clustering Method for Verb Frames and Ontology Acquisition,” *Proc. LREC-98 Workshop on Adapting Lexical and Corpus Resources to Sublanguages and Applications*, European Language Resources—Distribution Agency, Paris, 1998.
9. F. Esposito et al., “Learning from Parsed Sentences with INTHELEX,” *Proc. Learning Language in Logic Workshop (LLL-2000) and Learning Language in Logic Workshop (LLL-2000)*, Assoc. for Computational Linguistics, New Brunswick, N.J., 2000, pp. 194–198.
10. A. Maedche and S. Staab, “Discovering Conceptual Relations from Text,” *Proc. European Conf. Artificial Intelligence (ECAI-00)*, IOS Press, Amsterdam, 2000, pp. 321–325.
11. J.-U. Kietz, A. Maedche, and R. Volz, “Semi-Automatic Ontology Acquisition from a Corporate Intranet,” *Proc. Learning Language in Logic Workshop (LLL-2000)*, ACL, New Brunswick, N.J., 2000, pp. 31–43.
12. E. Morin, “Automatic Acquisition of Semantic Relations between Terms from Technical Corpora,” *Proc. of the Fifth Int’l Congress on Terminology and Knowledge Engineering (TKE-99)*, TermNet-Verlag, Vienna, 1999.
13. U. Hahn and K. Schnattinger, “Towards Text Knowledge Engineering,” *Proc. Am. Assoc. for Artificial Intelligence (AAAI-98)*, AAAI/MIT Press, Menlo Park, Calif., 1998.
14. M.A. Hearst, “Automatic Acquisition of Hyponyms from Large Text Corpora,” *Proc. Conf. Computational Linguistics (COLING-92)*, 1992.
15. Y. Wilks, B. Slator, and L. Guthrie, *Electric Words: Dictionaries, Computers, and Meanings*, MIT Press, Cambridge, Mass., 1996.
16. J. Jannink and G. Wiederhold, “Thesaurus Entry Extraction from an On-Line Dictionary,” *Proc. Second Int’l Conf. Information Fusion (Fusion-99)*, Omnipress, Wisconsin, 1999.
17. J.-U. Kietz and K. Morik, “A Polynomial Approach to the Constructive Induction of Structural Knowledge,” *Machine Learning*, vol. 14, no. 2, 1994, pp. 193–211.
18. S. Schlobach, “Assertional Mining in Description Logics,” *Proc. 2000 Int’l Workshop on Description Logics (DL-2000)*, 2000; <http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-33>.
19. A. Doan, P. Domingos, and A. Levy, “Learning Source Descriptions for Data Integration,” *Proc. Int’l Workshop on The Web and Databases (WebDB-2000)*, Springer-Verlag, Berlin, 2000, pp. 60–71.
20. P. Johannesson, “A Method for Transforming Relational Schemas into Conceptual Schemas,” *Proc. Int’l Conf. Data Engineering (IDCE-94)*, IEEE Press, Piscataway, N.J., 1994, pp. 190–201.
21. Z. Tari et al., “The Reengineering of Relational Databases Based on Key and Data Correlations,” *Proc. Seventh Conf. Database Semantics (DS-7)*, Chapman & Hall, 1998, pp. 40–52.

Table A. A survey of ontology-learning approaches.

Domain	Methods	Features used	Prime purpose	Papers
Free Text	Clustering	Syntax	Extract	Paul Buitelaar, ⁶ H. Assadi, ⁷ and David Faure and Claire Nedellec ⁸
	Inductive logic programming	Syntax, logic representation	Extract	
Free Text	Association rules	Syntax, Tokens	Extract	Alexander Maedche and Steffen Staab ¹⁰
	Frequency-based	Syntax	Prune	
	Pattern matching	—	Extract	
Free Text	Classification	Syntax, semantics	Refine	Joerg-Uwe Kietz et al. ¹¹
				Emanuelle Morin ¹²
Dictionary	Information extraction	Syntax	Extract	Marti Hearst, ¹⁴ Yorik Wilks, ¹⁵ and Joerg-Uwe Kietz et al. ¹¹
	Page rank	Tokens	—	
Knowledge base	Concept induction, A-Box mining	Relations	Extract	Jan Jannink and Gio Wiederhold ¹⁶
Semistructured schemata	Naive Bayes	Relations	Reverse engineering	Joerg-Uwe Kietz and Katharina Morik ¹⁷ and S. Schlobach ¹⁸
Relational schemata	Data correlation	Relations	Reverse engineering	Anahai Doan et al. ¹⁹
				Paul Johannesson ²⁰ and Zahir Tari et al. ²¹

Lexical entry and concept extraction

One of the baseline methods applied in our framework for acquiring lexical entries with corresponding concepts is lexical entry and concept extraction. Text-To-Onto processes Web documents on the morphological level, including multiword terms such as “database reverse engineering” by n-grams, a simple statistics-based technique. Based on this text pre-processing, we apply term-extraction techniques, which are based on (weighted) statistical frequencies, to propose new lexical entries for L .

Often, the ontology engineer follows the proposal by the lexical entry and concept-extraction mechanism and includes a new lexical entry in the ontology. Because the new lexical entry comes without an associated concept, the ontology engineer must then decide (possibly with help from further processing) whether to introduce a new concept or link the new lexical entry to an existing concept.

Hierarchical concept clustering

Given a lexicon and a set of concepts, one major next step is taxonomic concept classification. One generally applicable method with regard to this is hierarchical clustering, which exploits items’ similarities to propose a hierarchy of item categories. We compute the similarity measure on the properties of items.

When extracting a hierarchy from natural-language text, term adjacency or syntactical relationships between terms yield considerable descriptive power to induce the semantic hierarchy of concepts related to these terms.

David Faure and Claire Nedellec give a sophisticated example for hierarchical clustering (see the sidebar). They present a cooperative machine-learning system, Asium (acquisition of semantic knowledge using machine-learning method), which acquires taxonomic relations and subcategorization frames of verbs based on syntactic input. The Asium system hierarchically clusters nouns based on the verbs to which they are syntactically related and vice versa. Thus, they cooperatively extend the lexicon, the concept set, and the concept heterarchy (L , C , H_C).

Dictionary parsing

Machine-readable dictionaries are frequently available for many domains. Although their internal structure is mostly free text, comparatively few patterns are used to give text definitions. Hence, MRDs exhibit a large degree of regularity that can be exploited to extract a domain conceptualization.

We have used Text-To-Onto to generate a concept taxonomy from an insurance company’s MRD (see the sidebar). Likewise, we’ve applied morphological processing to term extraction from free text—this time, however, complementing several pattern-matching heuristics. Take, for example, the following dictionary entry:

Automatic Debit Transfer: Electronic service arising from a debit authorization of the Yellow Account holder for a recipient to debit bills that fall due direct from the account....

We applied several heuristics to the morphologically analyzed definitions. For instance, one simple heuristic relates the definition term, here *automatic debit transfer*, with

Targeting completeness for the domain model appears to be practically unmanageable and computationally intractable, but targeting the scarcest model overly limits expressiveness.

the first noun phrase in the definition, here *electronic service*. The heterarchy $H_C : H_C$ (automatic debit transfer, electronic service) links their corresponding concepts. Applying this heuristic iteratively, we can propose large parts of the target ontology—more precisely, L , C , and H_C to the ontology engineer. In fact, because verbs tend to be modeled as relations, we can also use this method to extend R (and the linkage between R and L).

Association rules

One typically uses association-rule-learning algorithms for prototypical applications of data mining—for example, finding associations that occur between items such as supermarket products in a set of transactions for example customers’ purchases. The generalized association-rule-learning algorithm extends its baseline by aiming at descriptions at the appropriate taxonomy level—for example, “snacks are purchased together with drinks,” rather than “chips are purchased with beer,” and “peanuts are purchased with soda.”

In Text-To-Onto (see the sidebar), we use a modified generalized association-rule-learning algorithm to discover relations between concepts. A given class hierarchy H_C serves as background knowledge. Pairs of syntactically related concepts—for example, pair (festival, island) describing the head-modifier relationship contained in the sentence “The festival on Usedom attracts tourists from all over the world.”—are given as input to the algorithm. The algorithm generates association rules that compare the relevance of different rules while climbing up or down the taxonomy. The algorithm proposes what appears to be the most relevant binary rules to the ontology engineer for modeling relations into the ontology, thus extending R .

As the algorithm tends to generate a high number of rules, we offer various interaction modes. For example, the ontology engineer can restrict the number of suggested relations by defining so-called restriction concepts that must participate in the extracted relations. The flexible enabling and disabling of taxonomic knowledge for extracting relations is another way of focusing.

Figure 4 shows various views of the results. We can induce a generalized relation from the example data given earlier—relation $rel(event, area)$, which the ontology engineer could name *locatedin*, namely, *events* located in an *area* (which extends L and G). The user can add extracted relations to the ontology by dragging and dropping them. To explore and determine the right aggregation level of adding a relation to the ontology, the user can browse the relation views for extracted properties (see the left side of Figure 4).

Pruning

A common theme of modeling in various disciplines is the balance between completeness and domain-model scarcity. Targeting completeness for the domain model appears to be practically unmanageable and computationally intractable, but targeting the scarcest model overly limits expressiveness. Hence, we aim for a balance between the two that works. Our model should capture a rich target-domain conceptualization but exclude the parts out of its focus. Ontology import and reuse as well as ontology extraction put the scale considerably out of balance where out-of-focus concepts reign. Therefore, we appropriately diminish the ontology in the pruning phase.

We can view the problem of pruning in at least two ways. First, we need to clarify how

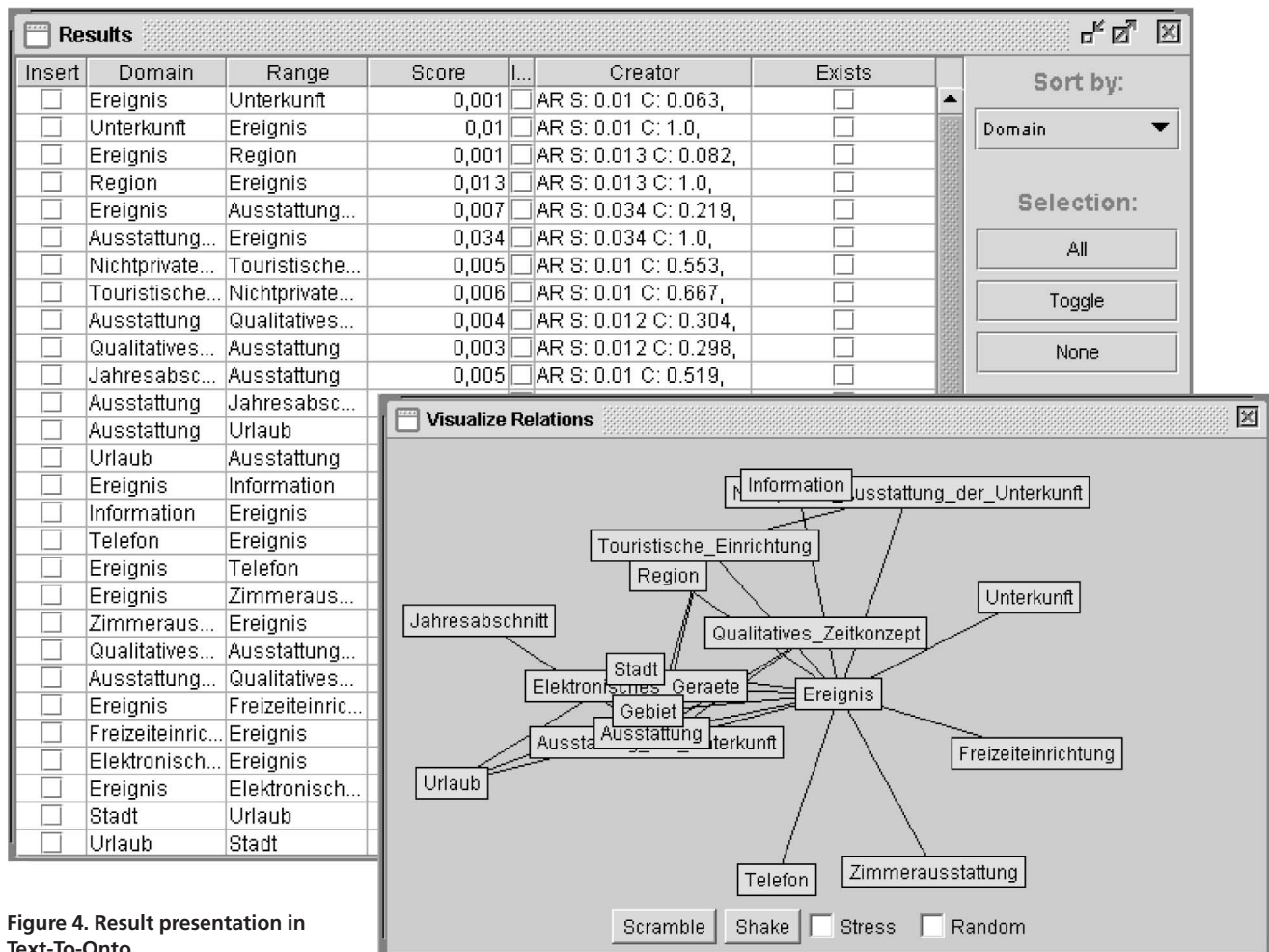


Figure 4. Result presentation in Text-To-Onto.

pruning particular parts of the ontology (for example, removing a concept or relation) affects the rest. For instance, Brian Peterson and his colleagues have described strategies that leave the user with a coherent ontology (that is, no dangling or broken links).⁶ Second, we can consider strategies for proposing ontology items that we should either keep or prune. Given a set of application-specific documents, several strategies exist for pruning the ontology that are based on absolute or relative counts of term frequency combined with the ontology's background knowledge (see the sidebar).

Refinement

Refining plays a similar role to extracting—the difference is on a sliding scale rather than a clear-cut distinction. Although

extracting serves mainly for cooperative modeling of the overall ontology (or at least of very significant chunks of it), the refinement phase is about fine-tuning the target ontology and the support of its evolving nature. The refinement phase can use data that comes from a concrete Semantic Web application—for example, log files of user queries or generic user data. Adapting and refining the ontology with respect to user requirements plays a major role in the application's acceptance and its further development.

In principle, we can use the same algorithms for extraction and refinement. However, during refinement, we must consider in detail the existing ontology and its existing connections, while extraction works more often than not practically from scratch.

Udo Hahn and Klemens Schnattinger presented a prototypical approach for refinement (see the sidebar)—although not for extraction! They introduced a methodology for automating the maintenance of domain-specific taxonomies. This incrementally updates an ontology as it acquires new concepts from text. The acquisition process is centered on the linguistic and conceptual “quality” of various forms of evidence underlying concept-hypothesis generation and refinement. Particularly, to determine a particular proposal's quality, Hahn and Schnattinger consider semantic conflicts and analogous semantic structures from the knowledge base for the ontology, thus extending an existing ontology with new lexical entries for L , new concepts for C , and new relations for H_C .

Ontology learning could add significant leverage to the Semantic Web because it propels the construction of domain ontologies, which the Semantic Web needs to succeed. We have presented a comprehensive framework for ontology learning that crosses the boundaries of single disciplines, touching on a number of challenges. The good news is, however, that you don't need perfect or optimal support for cooperative ontology modeling. At least according to our experience, cheap methods in an integrated environment can tremendously help the ontology engineer.

While a number of problems remain within individual disciplines, additional challenges arise that specifically pertain to applying ontology learning to the Semantic Web. With the use of XML-based namespace mechanisms, the notion of an ontology with well-defined boundaries—for example, only definitions that are in one file—will disappear. Rather, the Semantic Web might yield an amoeba-like structure regarding ontology boundaries because ontologies refer to and import each other (for example, the DAML-ONT primitive `import`). However, we do not yet know what the semantics of these structures will look like. In light of these facts, the importance of methods such as ontology pruning and crawling will drastically increase. Moreover, we have so far restricted our attention in ontology learning to the conceptual structures that are almost contained in RDF(S). Additional semantic layers on top of RDF (for example, future OIL or DAML-ONT with axioms, A) will require new means for improved ontology engineering with axioms, too! ■

Acknowledgments

We thank our students, Dirk Wenke, and Raphael Volz for work on *OntoEdit* and *Text-To-Onto*. Swiss Life/Rentananstalt (Zurich, Switzerland), Ontoprise GmbH (Karlsruhe, Germany), the US Air Force DARPA DAML ("OntoAgents" project), the European Union (IST-1999-10132 "On-To-Knowledge" project), and the German BMBF (01IN901C0 "GETESS" project) partly financed research for this article.

References

1. E. Grosso et al., "Knowledge Modeling at the Millennium—the Design and Evolution of Protégé-2000," *Proc. 12th Int'l Workshop Knowledge Acquisition, Modeling and Management (KAW-99)*, 1999.
2. G. Webb, J. Wells, and Z. Zheng, "An Experimental Evaluation of Integrating Machine Learning with Knowledge Acquisition," *Machine Learning*, vol. 35, no. 1, 1999, pp. 5–23.
3. K. Morik et al., *Knowledge Acquisition and Machine Learning: Theory, Methods, and Applications*, Academic Press, London, 1993.
4. B. Gaines and M. Shaw, "Integrated Knowledge Acquisition Architectures," *J. Intelligent Information Systems*, vol. 1, no. 1, 1992, pp. 9–34.
5. K. Morik, "Balanced Cooperative Modeling," *Machine Learning*, vol. 11, no. 1, 1993, pp. 217–235.
6. B. Peterson, W. Andersen, and J. Engel, "Knowledge Bus: Generating Application-Focused Databases from Large Ontologies," *Proc. Fifth Workshop Knowledge Representation Meets Databases (KRDB-98)*, 1998, <http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-10> (current 19 Mar. 2001).
7. S. Staab et al., "Knowledge Processes and Ontologies," *IEEE Intelligent Systems*, vol. 16, no. 1, Jan./Feb. 2001, pp. 26–34.
8. S. Staab and A. Maedche, "Knowledge Portals—Ontologies at Work," to be published in *AI Magazine*, vol. 21, no. 2, Summer 2001.
9. G. Miller, "WordNet: A Lexical Database for English," *Comm. ACM*, vol. 38, no. 11, Nov. 1995, pp. 39–41.
10. G. Neumann et al., "An Information Extraction Core System for Real World German Text Processing," *Proc. Fifth Conf. Applied Natural Language Processing (ANLP-97)*, 1997, pp. 208–215.
11. G. Stumme and A. Maedche, "FCA-Merge: A Bottom-Up Approach for Merging Ontologies," to be published in *Proc. 17th Int'l Joint Conf. Artificial Intelligence (IJCAI '01)*, Morgan Kaufmann, San Francisco, 2001.

The Authors



Alexander Maedche is a PhD student at the Institute of Applied Informatics and Formal Description Methods at the University of Karlsruhe. His research interests include knowledge discovery in data and text, ontology engineering, learning and application of ontologies, and the Semantic Web. He recently founded together with Rudi Studer a research group at the FZI Research Center for Information Technologies at the University of Karlsruhe that researches Semantic Web technologies and applies them to knowledge management applications in practice. He received a diploma in industrial engineering, majoring in computer science and operations research, from the University of Karlsruhe. Contact him at the Institute AIFB, Univ. of Karlsruhe, 76128 Karlsruhe, Germany; ama@aifb.uni-karlsruhe.de.



Steffen Staab is an assistant professor at the University of Karlsruhe and cofounder of Ontoprise GmbH. His research interests include computational linguistics, text mining, knowledge management, ontologies, and the Semantic Web. He received an MSE from the University of Pennsylvania and a Dr. rer. nat. from the University of Freiburg, both in informatics. He organized several national and international conferences and workshops, and is now chairing the Semantic Web Workshop in Hongkong at WWW10. Contact him at the Institute AIFB, Univ. of Karlsruhe, 76128 Karlsruhe, Germany; sst@aifb.uni-karlsruhe.de.

Ontologies in RDF(S)

Steffen Staab, Michael Erdmann, Alexander Mädche

Abstract

RDF(S)¹ constitutes a newly emerging standard for metadata that is about to turn the World Wide Web into a machine-understandable knowledge base. It is an XML application that allows for the denotation of facts and schemata in a web-compatible format, building on an elaborate object-model for describing concepts and relations. Thus, it might turn up as a natural choice for a widely-useable ontology description language. However, its lack of capabilities for describing the semantics of concepts and relations beyond those provided by inheritance mechanisms makes it a rather weak language for even the most austere knowledge-based system. This paper² presents an approach for modeling ontologies in RDF(S) that also considers axioms as objects that are describable in RDF(S). Thus, we provide flexible, extensible, and adequate means for accessing and exchanging axioms in RDF(S). Our approach follows the spirit of the World Wide Web, as we do not assume a global axiom specification language that is too intractable for one purpose and too weak for the next, but rather a methodology that allows (communities of) users to specify what axioms are interesting in their domain.

¹We use “RDF(S)” to refer to the combined technologies of RDF and RDFS.

²This paper is a revised and extended version of a paper presented at the ECDL-2000 Workshop on *Semantic Web*, Lisbon, Portugal, September 2000.

1 Introduction

The development of the World Wide Web is about to mature from a technical platform that allows for the transportation of information from sources to humans (albeit in many syntactic formats) to the communication of knowledge from Web sources to machines. The knowledge food chain has started with technical protocols and preliminary formats for information presentation (HTML – Hyper-Text Markup Language) over a general methodology for separating information contents from layout (XML – eXtensible Markup Language, XSL – eXtensible Stylesheet Language) to reach the realms of knowledge provisioning by the means of RDF and RDFS.

RDF (Resource Description Framework) is a W3C recommendation (Lassila & Swick, 1999) that provides description facilities for knowledge pieces, *viz.* for triples that denote relations between pairs of objects. To exchange and process RDF models they can be serialized in XML. RDF exploits the means of XML to allow for disjoint namespaces, linking and referring between namespaces and, hence, is a general methodology for sharing machine-processable knowledge in a distributed setting. On top of RDF the simple schema language *RDFS* (Resource Description Framework Schema; (Brickley & Guha, 1999)) has been defined to offer a distinguished vocabulary to model class and property hierarchies and other basic schema primitives that can be referred to from RDF models. To phrase the role of RDFS in knowledge engineering terminology, it allows to define a simple *ontology* that particular RDF documents may be checked against to determine consistency.

Ontologies have shown their usefulness in application areas such as intelligent information integration or information brokering. Therefore their use is highly interesting for web applications, which may also profit from long term experiences made in the knowledge acquisition community. Nevertheless, while support for modeling of ontological concepts and relations has been extensively provided in RDF(S), the same cannot be said about the modeling of ontological axioms — one of the key ingredients in ontology definitions and one of the major benefits of ontology applications.

RDF(S) offers only the most basic modeling primitives for ontology modeling. Even though there are good and bad choices for particular formal languages, one must face the principal trade-off between tractability and expressiveness of a language. RDF(S) has been placed nearer to the low end of expressiveness, because it has been conceived to be applicable to vast web resources! In contrast to common knowledge representation languages, RDF(S) has not been meant to be the definitive answer to all knowledge representation problems, but rather an *extensible core language*. The namespace and reification mechanisms of RDF(S) allow (communities of) users to define their very own standards in RDF(S) format — extending the core definitions and semantics. As RDF(S) leaves the well-trodden paths of knowledge engineering at this point, we must reconsider crucial issues concerning ontology modeling and ontology applications. To name but a few, we mention the problem of merging and mapping between namespaces, scalability issues, or the definition and usage of ontological axioms.

In this paper we concentrate on the latter, namely on how to model axioms in RDF(S) following the stipulations, *(i)*, that the core semantics of RDF(S) is reused such that “pure” RDF(S) applications may still process the core object-model definitions, *(ii)*, that the semantics is preserved between different inferencing tools (at least to a large extent), and, *(iii)*, that axiom modeling is adaptable to reflect diverging needs of different communities. Current proposals neglect or even conflict with one or several of these requirements. For instance, the first requirement is violated by the ontology exchange language XOL (Karp et al., 1999) making

all the object-model definitions indigestible for most RDF(S) applications. The interchangeability and adaptability stipulation is extremely difficult to meet by the parse-tree-based representation of MetaLog (Marchiori & Saarela, 1998), since it obliges to first-order logic formulae. We will show how to adapt a general methodology that we have proposed for axiom modeling (Staab & Maedche, 2000) to be applied to the engineering of ontologies with RDF(S). Our approach is based on translations of RDF(S) axiom specifications into various target systems that provide the inferencing services. As our running example, we map axiom specifications into an F-Logic format that has already served as the core system for SiLRi, an inference service for core RDF (Decker et al., 1998). Our methodology is centered around categorization of axioms, because this allows for a more concise description of the *semantic meaning* rather than a particular syntactic representation of axioms. Thus, we get a better grip on extensions and adaptations to particular target inferencing systems.

In the following, we introduce the RDF(S) data model and describe how to define an object model in RDF(S) including practical issues of ontology documentation (Section 2). Then we describe our methodology for using RDF(S) such that axioms may be engineered and exchanged. We describe the core idea of our approach and illustrate with several examples how to realize our approach (Section 3). In a case study (Section 4) we illustrate the application of our approach in our ontology engineering environment, *OntoEdit*, and our semantic community web portal, *KA2Portal* (Staab et al., 2000). Before we conclude, we give a brief survey of related work.

2 Modeling Concepts and Relations in RDF(S)

In this section we will first take a look at the core ontology engineering task, i.e. at the RDF(S) data model proper, and then exploit RDF(S) also for purposes of practical ontology engineering, viz. for documentation of newly defined or reused ontologies. This will lay the groundwork for the modeling of axioms in Section 3.

2.1 The RDF(S) Data Model

RDF(S) is an abstract data model that defines relationships between entities (called resources in RDF) in a similar fashion as semantic nets. Statements in RDF describe resources, that can be web pages or surrogates for real world objects like publications, pieces of art, persons, or institutions. We illustrate how concepts and relations can be modelled in RDF(S) by presenting a sample ontology in the abstract data model and only afterwards show how these concepts and relations are presented in the XML-serialisation of RDF(S).

2.1.1 RDF

As already mentioned RDF(S) consists of two closely related parts: RDF and RDF Schema. The foundation of RDF(S) is laid out by RDF which defines basic entities, like resources, properties, and statements. Anything in RDF(S) is a resource. Resources may be related to each other or to literal (i.e. atomic) values via properties. Such a relationship represents a statement that itself may be considered a resource, i.e. reification is directly built into the RDF data model. Thus, it is possible to make statements about statements. These basic notions can be easily depicted in a graphical notation that resembles semantic nets. To illustrate the possibilities of pure RDF the following statements are expressed in RDF and depicted

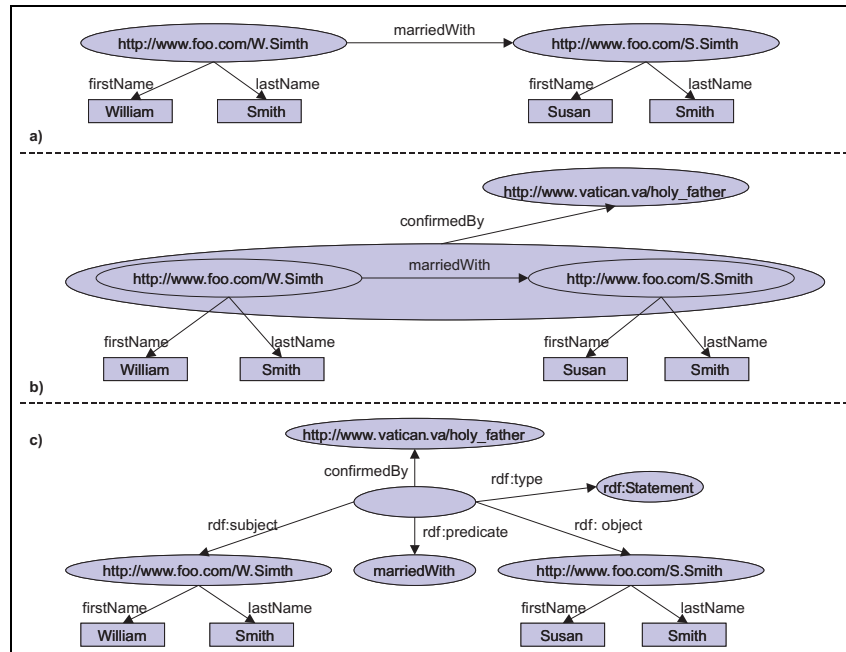


Figure 1: An example RDF data model.

in Figure 1³:

- Firstly, in part (a) of Figure 1 two resources are defined, each carrying a `FIRSTNAME` and a `LASTNAME` property with literal values, identifying the resources as William and Susan Smith, respectively. These two resources come with a URI as their unique global identifier and they are related via the property `MARRIEDWITH`, which expresses that William is married with Susan.
- Part (b) of the illustration shows a convenient shortcut for expressing more complex statements, i.e. reifying a statement and defining a property for the new resource. The example denotes that the marriage between William and Susan has been confirmed by the resource representing the Holy Father in Rome.
- The RDF data model offers the predefined resource `rdf:statement` and the predefined properties `rdf:subject`, `rdf:predicate`, and `rdf:object` to reify a statement as a resource. The actual model for the example (b) is depicted in part (c) of Figure 1. Note that the reified statement makes no claims about the truth value of what is reified, i.e. if one wants to express that William and Susan are married *and* that this marriage has been confirmed by the pope then the actual data model must contain a union of part (a) and part (c) of the example illustration.

2.1.2 RDFS

As a companion standard to RDF, the schema language RDFS is more important with respect to ontological modeling of domains. RDFS offers a distinguished vocabulary defined on top of RDF to allow the modelling of object models with

³Resources are represented by ovals, literal values by shaded rectangles and properties by directed, labeled arcs.

cleanly defined semantics. The terms introduced in RDFS build the groundwork for the extensions of RDF(S) that are proposed in this paper. The relevant RDFS terms are presented in the following list.

- The most general class in RDF(S) is `rdfs:Resource`. It has two subclasses, namely `rdfs:Class` and `rdf:Property` (cf. Figure 2⁴). When specifying a domain specific schema for RDF(S), the classes and properties defined in this schema will become instances of these two resources.
- The resource `rdfs:Class` denotes the set of all classes in an object-oriented sense. I.e. classes like `appl:Person` or `appl:Organisation` are instances of the meta-class `rdfs:Class`.
- The same holds for properties, i.e. each property defined in an application specific RDF schema is an instance of `rdf:Property`, e.g. `appl:marriedWith`
- RDFS defines the special property `rdfs:subClassOf` that defines the subclass relationship between classes. Since `rdfs:subClassOf` is transitive, definitions are inherited by the more specific classes from the more general classes and resources that are instances of a class are automatically instances of all superclasses of this class. In RDF(S) it is prohibited that any class is an `rdfs:subClassOf` itself or of one of its subclasses.
- Similar to `rdfs:subClassOf`, which defines a hierarchy of classes, another special type of relation `rdfs:subPropertyOf` defines a hierarchy of properties, e.g. one may express that `FATHEROF` is an `rdfs:subPropertyOf` `PARENTOF`.
- RDFS allows to define the domain and range restrictions associated with properties. For instance, these restrictions allow the definition that persons and only persons may be `MARRIEDWITH` and only with other persons.

As depicted in the middle layer of Figure 2 the domain specific classes `appl:Person`, `appl:Man`, and `appl:Woman` are defined as instances of `rdfs:Class`. In the same way domain specific property types are defined as instances of `rdf:Property`, i.e. `APPL:MARRIEDWITH`, `APPL:FIRSTNAME`, and `APPL:LASTNAME`.

2.1.3 The use of XML Namespaces in RDF(S)

The XML namespace mechanism plays a crucial role for the development of RDF schemata and applications. It allows to distinguish between different modeling layers (cf. Figure 2 and 3) and to reuse and integrate existing schemata and applications. At the time being, there exist a number of *canonical* namespaces, e.g. for RDF, RDFS, and Dublin Core (cf. Section 2.2). We here introduce two new namespaces that aim at two different objectives, viz. the comprehensive documentation of ontologies and the capturing of our proposal for the modeling of ontological axioms

An actual ontology definition occurs at a concrete URL⁵. It defines shorthand notations which refer to our actual namespaces for ontology documentation and modeling of ontological axioms, abbreviated `odoc` and `o`, respectively. An actual

⁴The reader may note that only a very small part of RDF(S) is depicted in the RDF/RDFS layer of the figure. Furthermore, the relation `APPL:MARRIEDWITH` in the data layer is identical to the resource `APPL:MARRIEDWITH` in the schema layer.

⁵The reader may actually compare with the documents that appear at these URLs, e.g. <http://ontoserver.aifb.uni-karlsruhe.de/schema/example.rdf>

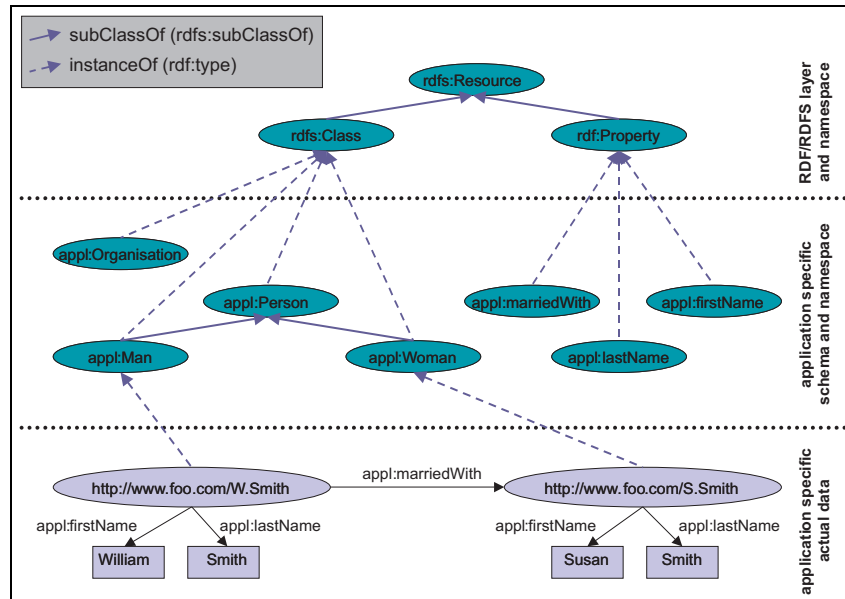


Figure 2: An example RDF schema and its embedding in RDF(S).

application that uses our example ontology will define a shorthand identifier like `appl` in order to refer to this particular, application-specific ontology. Figures 2, 3, and 4 presume these shorthand notations for the namespaces we have just mentioned.

2.1.4 XML serialization of RDF(S)

One important aspect for the success of RDF in the WWW is the way RDF models are represented and exchanged, namely via XML. In the following excerpt of the RDF schema document

`http://ontoserver.aifb.uni-karlsruhe.de/schema/example.rdf`, the classes and property types defined in Figure 2 are represented in XML and the domains and ranges of the properties are defined using the RDF constraint properties `rdfs:domain` and `rdfs:range`.

```
<rdf:Description ID="Person">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <rdfs:subClassOf
    rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
</rdf:Description>

<rdf:Description ID="Man">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <rdfs:subClassOf rdf:resource="#Person"/>
</rdf:Description>

<rdf:Description ID="Woman">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <rdfs:subClassOf rdf:resource="#Person"/>
</rdf:Description>

<rdf:Description ID="Organisation">
  <rdf:type resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Class"/>
  <rdfs:subClassOf
    rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource"/>
</rdf:Description>

<rdf:Description ID="firstName">
  <rdf:type
    resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Property"/>
  <rdfs:domain rdf:resource="#Person"/>
```



```

    <rdfs:range rdf:resource="http://www.w3.org/TR/xmlschema-2/#string"/>
  </rdf:Description>

  <rdf:Description ID="lastName">
    <rdf:type
      resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Property"/>
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="http://www.w3.org/TR/xmlschema-2/#string"/>
  </rdf:Description>

  <rdf:Description rdf:ID="marriedWith">
    <rdf:type
      resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Property"/>
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="#Person"/>
  </rdf:Description>

```

2.2 Modeling ontology metadata using RDF Dublin Core

Metadata about ontologies, such as the title, authors, version, statistical data, etc. are important for practical tasks of ontology engineering and exchange. In our approach we have adopted the well-established and standardized RDF Dublin Core Metadata element set (Weibel & Miller, 1998). This element set comprises fifteen elements which together capture basic aspects related to the description of resources. Ensuring a maximal level of generality and exchangeability, our ontologies are labeled using this basic element set. Since ontologies represent a very particular class of resource, the general Dublin Core metadata description does not offer sufficient support for ontology engineering and exchange. Hence, we describe further semantic types in the schema located at

<http://ontoserver.aifb.uni-karlsruhe.de/schema/ontodoc> and instantiate these types when we build a new ontology. The example below illustrates our usage and extension of Dublin Core by an excerpt of an exemplary ontology metadata description.

```

<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc = "http://purl.oclc.org/dc"
  xmlns:odoc = "http://ontoserver.aifb.uni-karlsruhe.de/schema/ontodoc">

  <rdf:Description about = "">
    <dc:Title>An Example Ontology</dc:Title>
    <dc:creator>
      <rdf:Bag>
        <rdf:li>Steffen Staab</rdf:li>
        <rdf:li>Michael Erdmann</rdf:li>
        <rdf:li>Alexander Maedche</rdf:li>
      </rdf:Bag>
    </dc:creator>
    <dc:date>2000-02-29</dc:date>
    <dc:format>text/xml</dc:format>
    <dc:description>
      An example ontology modeled for this small application
    </dc:description>
    <dc:subject>Ontology, RDF</dc:subject>

    <odoc:url>http://ontoserver.aifb.uni-karlsruhe.de/schema/example.rdf</odoc:url>
    <odoc:version>2.1</odoc:version>
    <odoc:last_modification>2000-03-01</odoc:last_modification>
    <odoc:ka_technique>semi-automatic text knowledge acquisition</odoc:ka_technique>
    <odoc:ontology_type>domain ontology</odoc:ontology_type>
    <odoc:no_concepts>24</odoc:no_concepts>
    <odoc:no_relations>13</odoc:no_relations>
    <odoc:no_axioms>9</odoc:no_axioms>
    <odoc:highest_depth_level>4</odoc:highest_depth_level>
  </rdf:Description>
</rdf:RDF>

```

3 Modeling of Axioms in RDF(S)

Having prepared the object-model and documentation backbone for ontologies in RDF(S), we may now approach the third pillar of our approach, viz. the specification of axioms in RDF(S). The basic idea that we pursue is the specification and serialization of axioms in RDF(S) such that they remain easily representable and exchangeable between different ontology engineering, representation and inferencing environments. The principal specification needs to be rather independent of particular target systems (to whatever extent this is possible at all) in order to be of value in a distributed web setting with many different basic applications.

3.1 Axioms are Objects, too

Representation of interesting axioms that are deemed to be applied in different inferencing applications turns out to be difficult. The reason is that typically some kind of non-propositional logic is involved that deals with quantifiers and quantifier scope. Axioms are difficult to grasp, since the representation of quantifier scope and its likes is usually what the nitty-gritty details of a particular syntax, on which a particular inferencing application is based, are about. An ontology representation in RDF(S) should, however, abstract from particular target systems.

A closer look at the bread and butter issues of ontology modeling reveals that many axioms that need to be formulated aim at much simpler purposes than arbitrary logic structures. Indeed, we have found that many axioms in our applications belong to one of a list of major axiom categories:

1. Axioms for a relational algebra
 - (a) Reflexivity of relations
 - (b) Irreflexivity of relations
 - (c) Symmetry of relations
 - (d) Asymmetry of relations
 - (e) Antisymmetry of relations
 - (f) Transitivity of relations
 - (g) Inverse relations
2. Composition of relations⁶
3. (Exhaustive) Partitions⁷
4. Axioms for subrelation relationships
5. Axioms that are derivations of the above mentioned
 - (a) Locally symmetric relations
 - (b) Locally transitive relations
 - (c) Locally inverse relations
6. Axioms for part-whole reasoning

⁶E.g., FATHERINLAWOF is composed by the relations FATHEROF and MARRIEDWITH.

⁷E.g., concepts *Man* and *Woman* share no instances.

Our principal idea for representing ontologies with axioms in RDF(S) is based on this categorization. The categories allow to distinguish between the structures that are repeatedly found in axiom specifications from a corresponding description in a particular language. Hence, one may describe axioms as complex objects (one could term them instantiations of axiom schemata) in RDF(S) that refer to concepts and relations, which are also denoted in RDF(S). For sets of axiom types we presume the definition of different RDF schemata. Similar to the case of simple metadata structures, the RDF schema responsible for an axiom categorization obliges to a particular semantics of its axiom types — which may be realized in a number of different inferencing systems like description logics systems (e.g., (Horrocks, 1998)) or frame logic systems (Decker et al., 1998). The schema defined in our namespace <http://ontoserver.aifb.uni-karlsruhe.de/schema/ontordf> stands for the semantics defined in this and our previous papers (Maedche et al., 2000; Staab & Maedche, 2000).⁸ The schema is also listed in the appendix of this paper (cf. Section A). Other communities may, of course, find other reasoning schemes more important, or they may just need an extension compared to what we provide here.

At the *symbol level*, we provide a RDF(S) syntax (i.e. serialization) to denote particular types of axioms. The categorization really constitutes a *knowledge level* that is independent from particular machines. In order to use an ontology denoted with our RDF(S) approach, one determines the appropriate axiom category and its actual instantiation found in a RDF(S) piece of ontology, translates it into a corresponding logical representation and executes it by an inferencing engine that is able to reason with (some of) the relevant axiom types.

Figure 3 summarizes our approach for modeling axiom specifications in RDF(S). It depicts the core of the RDF(S) definitions and our extension for axiom categorizations (i.e. our ontology meta layer). A simple ontology, especially a set of application specific relationships, is defined in terms of our extension to RDF(S). The basic motivation for our RDF(S) extension is to consider axioms as higher-order relations. Thereby, we must reflect the fact that properties in RDF(S) are binary relations and can only be used for moderately complex axiom specifications (e.g. symmetry of relations, but not composition). Axiom specifications that make use of n-ary relations require an (implicit) reification step — and, hence, are structured by the use of class representations with multiple associated properties.

In the following subsections, we will further elucidate our approach by proceeding through a few simple examples of our categorization of axiom specifications listed above. In particular our scheme is, (A) to show the representations of axioms in RDF(S) and (B) to show a structurally equivalent F(rame)-Logic representation that may easily be derived from its RDF(S) counterpart (cf. (Kifer et al., 1995; Decker, 1998; Lausen et al., 2001) on F-Logic). Then, (C) we exploit the expressiveness of F-Logic in order to specify translation axioms that work directly on the F-Logic object representation of axioms. Thus, (B) in combination with (C) describes a formally concise and executable translation. For better illustration, we finally, (D), indicate the result of our translation by exemplary target representations of the axioms stated in RDF(S).

The reader should note here that we do neither believe that F-Logic fulfills all the requirements that one might wish from an ontology inferencing language, nor do we believe that the axiom types we mention exhaust all relevant types. Rather we believe that our experiences in particular domains will push for further categorizations of axioms, further translation mechanisms, and, hence, further extensions of the core RDF(S) representation. All that will have to be agreed upon by communities that want to engineer and exchange ontologies with interesting axioms

⁸The reader may note that we have chosen names to coincide with many conventional names, e.g. “symmetry” of relations.

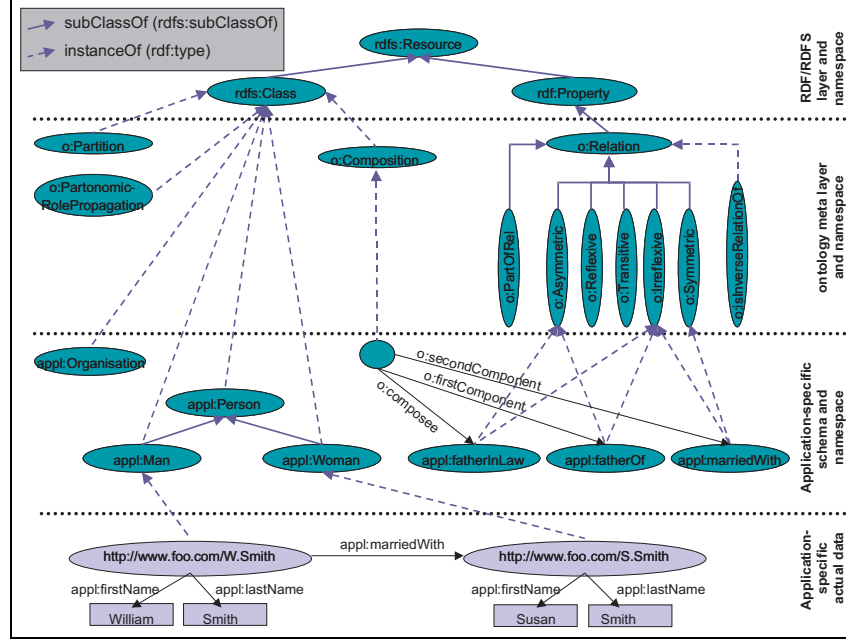


Figure 3: An excerpt of the example object model and an instantiation of classes, properties, and axioms in RDF(S)

across particularities of inference engines. Our main objective is to acquaint the reader with our *principle methodology* that is transportable to other translation approaches, inferencing systems, and other axiom types, when need arises.

3.2 Axioms for a relational algebra

The axiom types that we have shown above are listed such that simpler axioms tend to appear first. Axiom specifications that are referred to as “axioms for a relational algebra” rank among the simplest ones. They describe axioms with rather local effects, because their implications only affect one or two relations. We here show one simple example of these in order to explain the basic approach and some syntax. The principle approach easily transfers to all axiom types from 1.(a)-(g) to 6.

Let us consider an example for symmetry. A common denotation for the symmetry of a relation `MARRIEDWITH` (such as used for “William is married with Susan”) in first-order predicate logic boils down to:

$$(1) \quad \forall X, Y \quad \text{MARRIEDWITH}(X, Y) \leftarrow \text{MARRIEDWITH}(Y, X).$$

In F-Logic, this would be a valid axiom specification, too. Most often, however, modelers that use F-Logic take advantage of the object-oriented syntax. Concept definitions in F-Logic for *Person* having an attribute `MARRIEDWITH` and *Man* being a subconcept of *Person* is given in (2), while a fact that William is a *Man* who is `MARRIEDWITH` Susan appears like in (3).

$$(2) \quad \text{Person}[\text{MARRIEDWITH} \Rightarrow \text{Person}].$$

$$\text{Man}::\text{Person}.$$

$$(3) \quad \text{William}:\text{Man}[\text{MARRIEDWITH} \rightarrow \text{Susan}].$$

Hence, a rule corresponding to (1) is given by (4).

$$(4) \forall X, Y \quad Y[\text{MARRIEDWITH} \rightarrow X] \leftarrow X[\text{MARRIEDWITH} \rightarrow Y].$$

We denote symmetry as a predicate that holds for particular relations:

$$(5) \text{SYMMETRIC}(\text{MARRIEDWITH}).$$

In RDF(S), this specification may easily be realized by a newly agreed upon class `o:Symmetric`:

$$(6) <o:Symmetric \text{ rdf:ID="marriedWith"}/>$$

For a particular language like F-Logic, one may then derive the implications of symmetry by a general rule and, thus, ground the meaning of the predicate `SYMMETRIC` in a particular target system. The corresponding transformation rule (here in F-Logic) states that if for all symmetric relations R and object instances X and Y it holds that X is related to Y via R , then Y is also related to X via R .

$$(7) \forall R, X, Y \quad Y[R \rightarrow X] \leftarrow \text{SYMMETRIC}(R) \text{ and } X[R \rightarrow Y].$$

This small example already shows three advantages:

1. The axiom specification (6) is rather target-system independent.
2. It is easily realizable in RDF(S).
3. Our approach for denoting symmetry is much sparser than its initial counterpart (4), because (7) is implicitly assumed as the agreed semantics for our schema definition.

The latter point deserves some more attention: Obviously, the agreement cannot directly be described in RDF(S). Thus, it is subject to specifications of RDF(S) terminology *outside* of the RDF(S) documents, which need to be agreed upon by a community of users. This might be considered a disadvantage, however, this is eventually the case for RDF, RDFS, and all of their extensions like OIL or DAML+OIL.

Following our strategy sketched in the previous subsection, these steps from RDF representation to axiom meaning are now summarized in Table 1. For easier understanding, we will reuse this table layout also in the following subsection.

A	<code><o:Symmetric rdf:ID="marriedWith"/></code>	RDF(S)
B	<code>SYMMETRIC(MARRIEDWITH)</code>	F-Logic Representation
C	$\forall R, X, Y \quad Y[R \rightarrow X] \leftarrow \text{SYMMETRIC}(R)$ and $X[R \rightarrow Y].$	Translation Axiom
D	$\forall X, Y \quad X[\text{MARRIEDWITH} \rightarrow Y] \leftarrow$ $Y[\text{MARRIEDWITH} \rightarrow X].$	Target Axiom

Table 1: Symmetry

3.3 Composition of relations

The next example concerns composition of relations. For instance, if a first person is `FATHEROF` a second person who is `MARRIEDWITH` a third person then one may assert that the first person is the `FATHERINLAWOF` the third person. Again different inferencing systems may require completely different realizations of such an implication. The object description of such an axiom may easily be denoted in F-Logic or in RDF(S) (cf. Table 2). The transformation rule works very similarly as the transformation rule for symmetry.

A	<pre> <o:Composition rdf:ID="FatherInLawComp"> <o:composee rdf:Resource="fatherInLawOf"/> <o:firstComponent rdf:Resource="fatherOf"/> <o:secondComponent rdf:Resource="marriedWith"/> </o:Composition> </pre>
B	COMPOSITION(FATHERINLAWOF, FATHEROF, MARRIEDWITH)
C	$\forall R, Q, S, X, Y, Z \quad X[S \twoheadrightarrow Z] \leftarrow$ $\text{COMPOSITION}(S, R, Q) \text{ and } X[R \twoheadrightarrow Y] \text{ and } Y[Q \twoheadrightarrow Z].$
D	$\forall X, Y, Z \quad X[\text{FATHERINLAWOF} \twoheadrightarrow Z] \leftarrow$ $X[\text{FATHEROF} \twoheadrightarrow Y] \text{ and } Y[\text{MARRIEDWITH} \twoheadrightarrow Z].$

Table 2: Composition

3.4 Locally inverse relations

In our practice of implementing several knowledge portals⁹ we found that the inferences brought about by categories 1 to 4 were extremely useful. However, we also felt that these categories were often too general to be applied directly or when they were applied they easily yielded overly generic results.

For instance, given a conceptual model with concepts *WeddingParty*, *Person*, *PartyService* and properties HASMEMBER, PARTYAT, and SERVE. The definition of inverses is desirable. However, defining that the inverse of PARTYAT is HASMEMBER and that the inverse of SERVE is also HASMEMBER leads to undesired consequences, viz. it entails that for every person who does PARTYAT a particular wedding party there is a — correct — relation, HASMEMBER, from the party to this person and an — incorrect — relation, SERVE, from that person to that party. Thus, without intricate changes to the conceptual model one would have to live without the definition of inverses.

An easily viable way around the problem is the definition of *local inverseness* that also consider the domain of a relation¹⁰, before asserting an inverse relationship. Thus, (8) asserts that the inverse of HASMEMBER is PARTYAT, only if the corresponding domain is restricted to *WeddingParty*.

(8) LOCALINVERSE(*WeddingParty*, HASMEMBER, *Person*, PARTYAT)

Table 3 captures the corresponding translation and target axioms.

3.5 Part-Whole Reasoning

Commonly two versions of part-whole reasoning are distinguished and well-used.¹¹ First, *partonomic role propagation* is about propagating particular property values from parts to wholes. For instance, if the engine of my car is defunct, the whole car is defunct (“defunct” being propagated from the part “engine” to the whole “car”). However, if the rear window is broken, it might be less safe to drive the car, but it would be strange to consider it defunct.

Second, there is *concept specialization*. For instance, we might talk about a car SHOWING *wear-out-of-the-car* and about the *car-engine* SHOWING *wear-out-of-the-car-engine*. By using the means of concept specialization and

⁹Cf. (Staab et al., 2000; Staab & Maedche, 2001; Staab et al., 2001) and <http://ontobroker.aifb.uni-karlsruhe.de/demos.html>, <http://ontobroker.semanticweb.org/>.

¹⁰Actually, we have extended the model to consider either restrictions of the domain of a property or of its range or of its domain and its range.

¹¹Their usage is well accepted in medical informatics, whereas in general there is still a vivid discussion about the applicability of particular types of part-whole reasoning (cf., e.g., (Artale et al., 1996)).

A	<pre> <:LocalInverse rdf:ID="hasMemberVsPartyAt"> <:firstDomain rdf:Resource="WeddingParty"/> <:firstRelation rdf:Resource="hasMember"/> <:secondDomain rdf:Resource="Person"/> <:secondRelation rdf:Resource="partyAt"/> </:LocalInverse> </pre>
B	LOCALINVERSE(<i>WeddingParty</i> , HASMEMBER, <i>Person</i> , PARTYAT)
C	$\forall X, Y, R, C, Q, D \quad Y[Q \rightarrow X] \leftarrow$ $\text{LOCALINVERSE}(C, R, D, Q) \text{ and } X[R \rightarrow Y] \text{ and } X : C.$ $\forall X, Y, R, C, Q, D \quad Y[R \rightarrow X] \leftarrow$ $\text{LOCALINVERSE}(C, R, D, Q) \text{ and } X[Q \rightarrow Y] \text{ and } X : D.$
D	$\forall X, Y \quad Y[\text{PARTYAT} \rightarrow X] \leftarrow$ $X[\text{HASMEMBER} \rightarrow Y] \text{ and } X : \text{WeddingParty}.$ $\forall X, Y \quad Y[\text{HASMEMBER} \rightarrow X] \leftarrow X[\text{PARTYAT} \rightarrow Y] \text{ and } X : \text{Person}.$

Table 3: Locally inverse

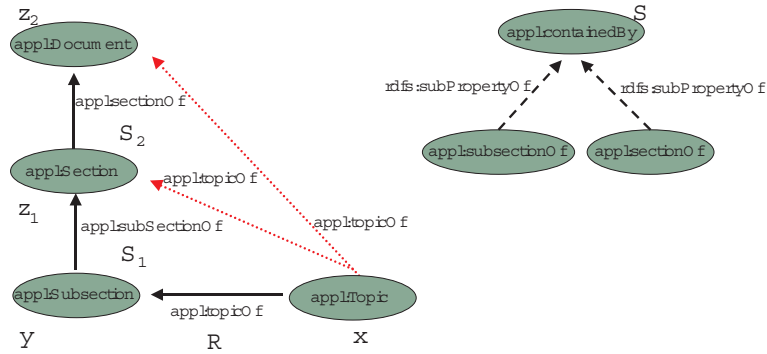


Figure 4: Partonomic Role Propagation

knowing that the *car-engine* is a part of the *car*, we may deduce that *wear-out-of-the-car-engine* is a subconcept of *wear-out-of-the-car*.

Because it is our main goal to show the principle of our approach, we only describe partonomic role propagation in order to illustrate how our approach may be applied to more complex axiom(combination)s.

Figure 4 depicts an example that propagates topics from subparts of documents to superparts. It takes four input parameters:

1. The relation that is propagated (TOPICOF in Figure 4), because not every relation is propagated from a part to a whole.
2. The relation that is propagating (CONTAINEDBY in Figure 4), because not every relation that may be propagated is propagated along all part-whole relations.
3. The whole up to which the relation is maximally propagated (*Document* in Figure 4), because propagation may be stopped (e.g. TOPICOF may be considered to be not propagated to an additional *Library*).
4. The concept for the instances of which the relation may be propagated (e.g., *Topic* in Figure 4), because not every class is treated the same (cf. (Hahn et al., 1999) for comprehensive examples).

A	<code><o:PartonomicRolePropagation rdf:ID="DocumentTopic"></code> <code> <o:propagatedrel rdf:Resource="topicOf"/></code> <code> <o:propagatingrel rdf:Resource="containedBy"/></code> <code> <o:topConcept rdf:Resource="Document"/></code> <code> <o:propConcept rdf:Resource="Topic"/></code> <code></o:PartonomicRolePropagation></code>
B	<code>PARTONOMICROLEPROPAGATION(TOPICOF, CONTAINEDBY, Document, Topic)</code>
C α	$\forall x, y, z, R, S, C, D \ x[R \twoheadrightarrow z] \leftarrow$ $\text{PARTONOMICROLEPROPAGATION}(R, S, C, D) \text{ and }$ $x : D \text{ and } x[R \twoheadrightarrow y] \text{ and } y[S \twoheadrightarrow z] \text{ and }$ $\text{PARTINSTANCEOF}(z, C, S).$
C β	$\forall z, C, S \ \text{PARTINSTANCEOF}(z, C, S) \leftarrow$ $\exists E \ z : E \text{ and } \text{PARTOFALONG}(E, C, S).$
C γ	$\forall C, S \ \text{PARTOFALONG}(C, C, S).$
C δ	$\forall E, C, S \ \text{PARTOFALONG}(E, C, S) \leftarrow$ $\exists F \ \text{PARTOFALONG}(E, F, S) \text{ and }$ $\exists Q \ F[Q \Rightarrow C] \text{ and } Q :: S.$
C ϵ	$\forall x, y, S, R \ x[S \twoheadrightarrow y] \leftarrow R :: S \text{ and } x[R \twoheadrightarrow y].$
D α	$\forall x, y, z \ x[\text{TOPICOF} \twoheadrightarrow z] \leftarrow$ $x : \text{Topic} \text{ and } x[\text{TOPICOF} \twoheadrightarrow y] \text{ and }$ $y[\text{CONTAINEDBY} \twoheadrightarrow z] \text{ and }$ $\text{PARTINSTANCEOF}(z, \text{Document}, \text{CONTAINEDBY}).$
D β	$\forall z \ \text{PARTINSTANCEOF}(z, \text{Document}, \text{CONTAINEDBY}) \leftarrow$ $\exists E \ z : E \text{ and } \text{PARTOFALONG}(E, \text{Document}, \text{CONTAINEDBY}).$
D γ	$\text{PARTOFALONG}(\text{Document}, \text{Document}, \text{CONTAINEDBY}).$
D δ	$\forall E, C \ \text{PARTOFALONG}(E, C, \text{CONTAINEDBY}) \leftarrow$ $\exists F \ \text{PARTOFALONG}(E, F, \text{CONTAINEDBY}) \text{ and }$ $\exists Q \ F[Q \Rightarrow C] \text{ and } Q :: \text{CONTAINEDBY}.$
D ϵ	$\forall x, y \ x[\text{CONTAINEDBY} \twoheadrightarrow y] \leftarrow$ $(x[\text{SUBSECTIONOF} \twoheadrightarrow y] \text{ or } x[\text{SECTIONOF} \twoheadrightarrow y]).$
D ϕ	$\text{SECTIONOF} :: \text{CONTAINEDBY}. \ \text{SUBSECTIONOF} :: \text{CONTAINEDBY}.$

Table 4: Partonomic Role Propagation

Table 4 captures the corresponding representations and translations. The RDF(S) part only captures the principal structure referring to the four main parameters as just given. A trivial corresponding F-Logic translation is given in Table 4B. Table 4C $\alpha - \epsilon$ takes as its input the principal structure and adds some axioms to do the actual propagation if sufficient conditions are fulfilled (C α). It needs a specification of which instances are subparts of *Document* along subrelations of the propagating relation (e.g., *CONTAINEDBY*). This is done by C $\beta - \delta$. For the inferencing between relations and subrelations in this context, it needs C ϵ . Table 4 shows the instantiation for our actual example. One may recognize that even in this small, concrete example there are still many inferences that may be drawn in D $\alpha - \epsilon$.

3.6 General axioms

Our approach of axiom categorization is not suited to cover every single axiom specification one may think of. Hence, we still must allow for axioms that are specified in a particular language like first-order predicate logic and we must allow for their representation in RDF(S). There are principally two ways to approach this problem. First, one may conceive a new RDF(S) representation format that is dedicated to a particular inferencing system for reading and performing inferences.

This is the way that has been chosen for OIL (Broekstra et al., 2001), which has a RDF(S) style representation for a core description logics, or Metalog (Marchiori & Saarela, 1998), which represents Horn clauses in RDF(S) format.

The alternative is to fall back to a representation that is even more application specific, viz. the encoding of ontological axioms in pure text, or “CDATA” in RDF speak (cf. the example below). In fact, the latter is a very practical choice for many application-specific axioms — once you make very deep assumptions about a particular representation, you are also free to use whatever format you like.

Our example here takes as input the current relationship status and implies who was the groom, the bride and the one who paid the bill (here it is the bride’s father).

```
<o:GeneralAxiom rdf:ID="WhoPaidForTheWeddingParty">
  <o:text lang="flogic">
    <![CDATA[
      FORALL w, x, y, z
        w:Wedding[groom->x, bride->y, billTo->z] <-
          z[fatherInLawOf->x:Man] AND x[marriedWith->y].
    ]]>
  </o:text>
</o:GeneralAxiom>
```

4 Case Study: Semantic Community Web Portal

4.1 Setting

The semantic community Web portal, KA2Portal, that we have built for the knowledge acquisition community (Staab et al., 2000) serves as an entry point for linking to and sharing knowledge about the knowledge acquisition research community and its work. Information may be provided and accessed using ontology structures for easier discovery of resources as well as for easier development and maintenance of the portal.

Figure 5 depicts the overall framework. Based on the community ontology, actual facts are contributed by the knowledge acquisition community or crawled from their web sites, e.g. annotated web pages or RDF sources. The SiLRi reasoning engine (Decker et al., 1998) provides the ontology service and allows for querying the accumulated knowledge base. Thus it acts as the back end for the actual portal. Users may either query the inference engine by clicking together a query or they may use forms of the web portal to retrieve knowledge.

Semantic inferences are crucial for the service that is provided by the KA2Portal. So far, people could contribute semantic information, e.g. by providing RDF sources. However, it was so far impossible to provide semantics about these RDF sources beyond the means that are inherent in core RDF(S). Conversely, the ontology was described in F-Logic (Kifer et al., 1995). Therefore, people could look at the KA2 ontology, but the ontology was initially not built with the spirit of the Semantic Web, as it was not transparent to software agents on the Web and not reusable by them.

Hence, there came up the need for representing the whole ontology, including axioms, on the Web. Thereby we felt two needs. First we did not want to come up with our own idiosyncratic syntax, but rather wanted to adhere to RDF(S) mechanisms as elaborated above. Second, we wanted to provide some tool support for engineering axioms. The framework that we have shown above allowed us both. We will show some excerpt of its application in the KA2Portal in the following.

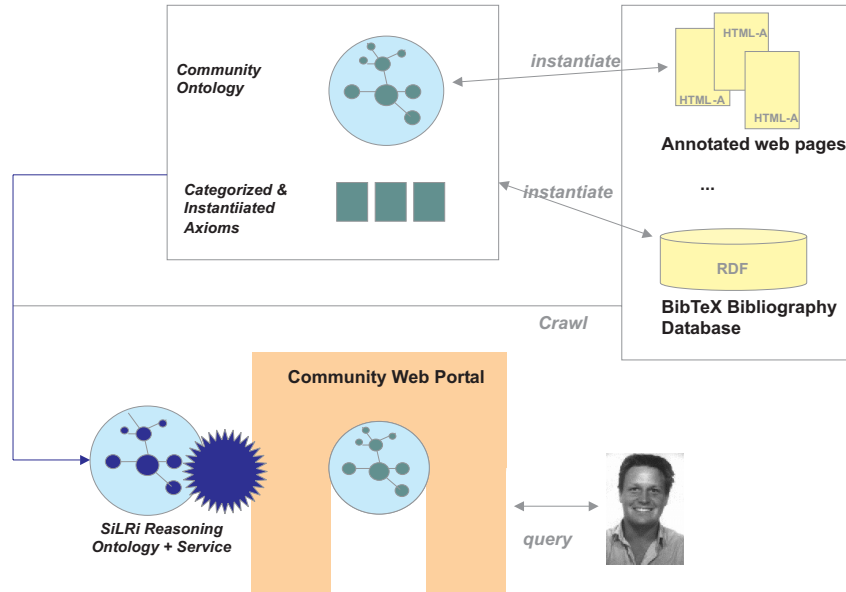


Figure 5: KA2Portal

4.2 Modeling the core ontology

We use our Ontology Engineering Environment OntoEdit for engineering class and property definitions in RDF(S) with graphical means. In particular, we may express that (cf. Figure 6 left and upper right window)

- *Book*, *Journal*, and *Article* are all subclasses of *Publication*
- A *Book* CONTAINSARTICLE *Article*
- A *Journal* CONTAINSARTICLE *Article*
- An *Article* is INBOOK *Book*
- An *Article* is INJOURNAL *Journal*

This is specified in RDF(S) as follows:

```
<rdfs:Class rdf:ID="Publication"/>
<rdfs:Class rdf:ID="Article">
  <rdfs:subClassOf rdf:resource="Publication"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Book">
  <rdfs:subClassOf rdf:resource="Publication"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Journal">
  <rdfs:subClassOf rdf:resource="Publication"/>
</rdfs:Class>

<rdf:Property rdf:ID="containsArticle">
  <rdfs:domain rdf:resource="Publication"/>
  <rdfs:range rdf:resource="Article"/>
</rdf:Property>
<rdf:Property rdf:ID="inBook">
  <rdfs:domain rdf:resource="Article"/>
  <rdfs:range rdf:resource="Book"/>
</rdf:Property>
<rdf:Property rdf:ID="inJournal"/>
```

```

<rdfs:domain rdf:resource="Article"/>
<rdfs:range rdf:resource="Journal"/>
</rdf:Property>

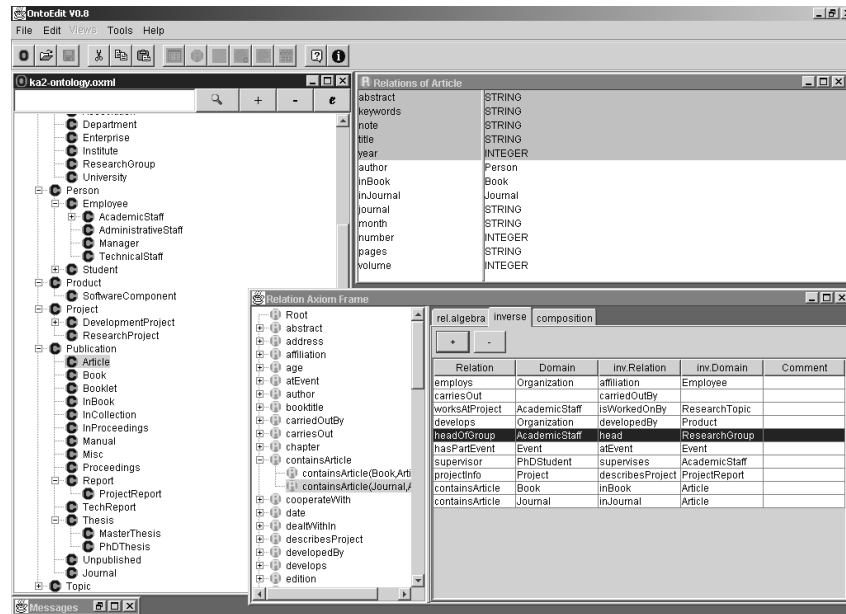
```

The ontology defines the conceptual backbone for contributing RDF metadata, e.g. bibliography entries:

```

<ka2:Book rdf:ID="book:WeavingTheWeb">
  <ka2:author rdf:resource="http://w3c.org/person/tbl"/>
  <ka2:title>Weaving The Web</ka2:title>
  ...
</ka2:Book>

```



Left: KA2 Taxonomy; Right: Interface for Inverseness and Local Inverseness of Relations

Figure 6: Snapshot of OntoEdit Web Ontology Workbench

4.3 Modeling an ontology with axioms in RDF(S)

The axioms complete the core ontology. For KA2Portal we use locally inverse relations in order to express that:

- If a particular book CONTAINSARTICLE a particular article, this article is related via INBOOK to that book.
- If a particular article appears INBOOK in a particular book, this book is related via CONTAINSARTICLE that article.
- If a particular journal CONTAINSARTICLE a particular article, this article is related via INJOURNAL to that book.
- If a particular article appears INJOURNAL in a particular journal, this journal is related via CONTAINSARTICLE that article.

Specified in our RDF(S) style denotation of axioms this boils down to:

```
<o:LocalInverse>
  <o:firstDomain rdf:Resource="Book"/>
  <o:firstRelation rdf:Resource="containsArticle"/>
  <o:secondDomain rdf:Resource="Article"/>
  <o:secondRelation rdf:Resource="inBook"/>
</o:LocalInverse>
<o:LocalInverse>
  <o:firstDomain rdf:Resource="Journal"/>
  <o:firstRelation rdf:Resource="containsArticle"/>
  <o:secondDomain rdf:Resource="Article"/>
  <o:secondRelation rdf:Resource="inJournal"/>
</o:LocalInverse>
```

OntoEdit allows to provide the components, `o:firstDomain`, `o:firstRelation`, `o:secondDomain`, `o:secondRelation`, in a table (cf. last two lines in table of right lower window in Figure 6). If `o:firstDomain` and `o:secondDomain` are provided, OntoEdit outputs the specification for locally inverse relations. If only `o:firstRelation` and `o:secondRelation` are provided, they are assumed to be immediate inverses.

Our approach allows for very easy graphical representation of axioms. For practical purposes the graphical representation proved extremely beneficial. System builders who may be experienced Java programmers, but who are quite often novices in logics are freed from syntactical errors (though of course not from semantic ones). The system builders we worked with — who were students in industrial engineering — were much more willing to deal with the GUI representation rather than the RDF or F-Logic syntax.

5 Related Work

The proposal described in this paper is based on several related approaches, viz. we have built on considerations made for the RDF inference service SiLRi (Decker et al., 1998), the ontology engineering environments ODE (Blázquez et al., 1998) and Protégé (Grosso et al., 1999), the ontology interchange language OIL (Broekstra et al., 2001), considerations made by Gruber (Gruber, 1993), and our own earlier work on general ontology engineering (Maedche et al., 2000; Staab & Maedche, 2000).

A purpose similar to our general goal of representing ontologies in RDF(S) is pursued with OIL (Broekstra et al., 2001). Actually, OIL might be considered a very sophisticated instantiation of our approach, as the definition of concepts and relations in description logics is equivalent to the instantiation of a small number of axiom schemata in a particular logical framework (cf. (Brachman, 1979)).

The reader may note that RDF(S) is a very weak language — it does not even have Turing power unlike more general frameworks like first order predicate logics. Naturally, it is not possible to describe the semantics of a more powerful language like F-Logic in simple RDF(S). However, what is desirable and what was pursued by OIL as well as by our approach is to reuse RDF(S) semantics where possible and to add on top where RDF(S) alone is insufficient.

The principal question that arises here is whether one particular language on top of RDF(S) is a single solution. The answer is difficult to give. There are so many different semantic options for languages that it is hard to believe that anytime soon the world will agree on one particular language, i.e. construct a language top down that is efficient to handle and powerful enough for all widely different needs. On the other hand, it is much easier for a community to talk about some reasoning patterns important for them (e.g., partonomic role propagation in medicine).

Then communities may come up with many languages that cover some sets of reasoning patterns relevant in some communities, i.e. constructing many different languages bottom up starting from community needs. This second way is the one we advertise.

There are a number of other approaches for ontology exchange and representation in XML formats that we do not want to elaborate here, as they did not intend to support the RDF(S) metadata standard, which is one of our primary concerns (e.g. (Marchiori & Saarela, 1998; Karp et al., 1999; Heflin & Hendler, 2000)).

Concerning inferencing rather than representation, SiLRi (Decker et al., 1998) was one of the first approaches to provide inferencing facilities for RDF. It delivers most of the basic inferencing functions one wants to have in RDF and, hence, has provided a good start for many RDF applications. In fact, it even allows to use axioms, but these axioms may not be denoted in RDF, but only directly in F-Logic. It lacks capabilities for axiom representation in RDF(S) that our proposal provides.

Concerning engineering, we have discussed how to push the engineering of ontological axioms from the *symbol level* onto the *knowledge level* in our earlier proposals (Maedche et al., 2000; Staab & Maedche, 2000). There we follow and extend the general arguments made for ODE (Blázquez et al., 1998) and Ontolingua (Fikes et al., 1997). This strategy has helped us here in providing an RDF(S) object representation for a number of different axiom types. Nearest to our actual RDF(S)-based ontology engineering tool is Protégé (Grosso et al., 1999), which provides comprehensive and sophisticated support for editing RDFS and RDF. Nevertheless, Protégé currently lacks any support for axiom modeling and inferencing — though our approach may be very easy to transfer to Protégé, too.

6 Conclusion

We have presented a new approach towards engineering ontologies in RDF and RDFS. Our objectives aim at the usage of existing inferencing services such as provided by deductive database mechanisms (Decker et al., 1998) or description logics systems (Horrocks, 1998). We reach these objectives through a *methodology* that classifies axioms into axiom types according to their *semantic meaning*. Each type receives an object representation that abstracts from scoping issues and is easily representable in RDF(S). Axiom descriptions only keep references to concepts and relations necessary to distinguish one particular axiom of one type from another one of the same type.

Our proposed extension of RDF(S) has been made with a clear goal in mind — the complete retention of the expressibility and semantics of RDF(S) for the representation of ontologies. This includes the relationship between ontologies and instances, both represented in RDF(S). Especially, the notion of *consistency* (cf. (Brickley & Guha, 1999)) between an RDF model and a schema also holds for ontologies expressed in RDF(S). The integration of the newly defined resources has been carried out in a such a way that all RDF processors capable of processing RDF schemas can correctly interpret RDF models following the ontology schema, even if they do not *understand* the semantics of the resources in the \circ -namespace.

Special applications like OntoEdit (Maedche et al., 2000) can interpret the \circ -namespace correctly and thus fully benefit from the richer modelling primitives, if the RDF model is valid¹² according to the defined ontology schema.

Acknowledgements. The research presented in this paper has been partially funded by BMBF under grant number 01IN802 (project “GETESS”) and by EU in

¹²cf. the “Validator” section in <http://www.ics.forth.gr/proj/isst/RDF/> for a set of operations to check for validity

the IST-1999-10132 project On-To-Knowledge. We thank our student Dirk Wenke who implemented large parts of the RDF(S)-based ontology editor and our colleague Stefan Decker who first proposed RDF(S) for specification of ontologies. We also thank two anonymous reviewers whose comments have greatly improved the paper.

References

- Artale, A., Franconi, E., Guarino, N., & Pazzi, L. (1996). Part-whole relations in object-centered systems: An overview. *Data and Knowledge Engineering*, 20(3):347–383.
- Blázquez, M., Fernández, M., García-Pinar, J. M., & Gómez-Pérez, A. (1998). Building ontologies at the knowledge level using the ontology design environment. In *Proc. of the 11th Int. Workshop on Knowledge Acquisition, Modeling and Mangement (KAW'98), Banff, Canada, October 1998*.
- Brachman, R. (1979). On the epistemological status of semantic networks. *Associative Networks*, pages 3–50.
- Brickley, D. & Guha, R. (1999). Resource description framework (RDF) schema specification. Technical report, W3C. W3C Proposed Recommendation. <http://www.w3.org/TR/PR-rdf-schema/>.
- Broekstra, J., Klein, M., Decker, S., Fensel, D., van Harmelen, F., & Horrocks, I. (2001). Enabling knowledge representation on the web by extending RDF schema. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 467–478. ACM.
- Decker, S. (1998). On domain-specific declarative knowledge representation and database languages. In *Proc. of the 5th Knowledge Representation meets Databases Workshop (KRDB98)*, pages 9.1–9.7.
- Decker, S., Brickley, D., Saarela, J., & Angele, J. (1998). A query and inference service for RDF. In *QL'98 - The Query Languages Workshop*. W3C. <http://www.w3.org/TandS/QL/QL98/>.
- Fikes, R., Farquhar, A., & Rice, J. (1997). Tools for assembling modular ontologies in Ontolingua. In *Proc. of AAAI 97*, pages 436–441.
- Grosso, E., Eriksson, H., Fergerson, R. W., Tu, S. W., & Musen, M. M. (1999). Knowledge modeling at the millennium — the design and evolution of Protégé-2000. In *Proc. of the 12th International Workshop on Knowledge Acquisition, Modeling and Mangement (KAW'99), Banff, Canada, October 1999*.
- Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(1):199–220.
- Hahn, U., Schulz, S., & Romacker, M. (1999). Partonomic reasoning as taxonomic reasoning in medicine. In *Proc. of AAAI-99*, pages 271–276.
- Heflin, J. & Hendler, J. (2000). Dynamic Ontologies on the Web. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI-2000), Menlo Park, California, AAAI Press*, pages 443–449.
- Horrocks, I. (1998). Using an expressive description logic: FaCT or fiction? In *Proc. of KR-98*, pages 636–647.
- Karp, P. D., Chaudhri, V. K., & Thomere, J. (1999). XOL: An XML-based ontology exchange language. Technical report. Version 0.3.
- Kifer, M., Lausen, G., & Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843.
- Lassila, O. & Swick, R. R. (1999). Resource description framework (RDF) model and syntax specification. Technical report, W3C. W3C Recommendation. <http://www.w3.org/TR/REC-rdf-syntax>.

- Lausen, G., May, W., Frohn, J., Himmeröder, R., Kandzia, P., Schlepphorst, C., & Uphoff, H. (2001). Florid tutorial. Technical report, Institute for Informatics, University of Freiburg. <http://www.informatik.uni-freiburg.de/~dbis/florid/>.
- Maedche, A., Schnurr, H.-P., Staab, S., & Studer, R. (2000). Representation language-neutral modeling of ontologies. In Frank, U. (Ed.), *Proceedings of the German Workshop "Modellierung-2000"*. Koblenz, Germany, April, 5-7, 2000, pages 129–142. Fölbach-Verlag.
- Marchiori, M. & Saarela, J. (1998). Query + metadata + logic = metalog. In *QL'98 - The Query Languages Workshop*. W3C. <http://www.w3.org/TandS/QL/QL98/>.
- Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H.-P., Studer, R., & Sure, Y. (2000). Semantic community web portals. *Computer Networks*, 33(1-6):473–491. WWW9 — Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, May, 15-19, 2000.
- Staab, S. & Maedche, A. (2001). Knowledge portals — ontologies at work. *AI Magazine*, 21(2):63–75.
- Staab, S., Schnurr, H.-P., Studer, R., & Sure, Y. (2001). Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1):26–35.
- Staab, S. & Maedche, A. (2000). Axioms are objects, too — ontology engineering beyond the modeling of concepts and relations. In Benjamins, V., Gomez-Perez, A., & Guarino, N. (Eds.), *Proceedings of the ECAI 2000 Workshop on Ontologies and Problem-Solving Methods*. Berlin, August 21-22, 2000, pages 8.1–8.6.
- Weibel, S. & Miller, E. (1998). Dublin core metadata. Technical report. <http://purl.oclc.org/dc>.

A The RDF schema for categories of relationships

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

  <rdfs:Class ID="Relation">
    <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  </rdfs:Class>

  <rdfs:Class ID="Asymmetric">
    <rdfs:subClassOf rdf:resource="#Relation"/>
  </rdfs:Class>

  <rdfs:Class ID="Reflexive">
    <rdfs:subClassOf rdf:resource="#Relation"/>
  </rdfs:Class>

  <rdfs:Class ID="Transitive">
    <rdfs:subClassOf rdf:resource="#Relation"/>
  </rdfs:Class>

  <rdfs:Class ID="Irreflexive">
    <rdfs:subClassOf rdf:resource="#Relation"/>
  </rdfs:Class>

  <rdfs:Class ID="Symmetric">
    <rdfs:subClassOf rdf:resource="#Relation"/>
  </rdfs:Class>

  <rdfs:Class ID="PartOfRel">
    <rdfs:subClassOf rdf:resource="#Relation"/>
  </rdfs:Class>

  <rdf:Description ID="isInverseRelationOf">
    <rdf:type rdf:resource="#Relation"/>
  </rdf:Description>

  <!-- Definitions for LOCALLY-INVERSE-RELATIONS -->

  <rdfs:Class ID="LocalInverse"/>
```

```

<rdf:Property ID="firstDomain">
  <rdfs:domain rdf:resource="#LocalInverse"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Class"/>
</rdf:Property>

<rdf:Property ID="firstRelation">
  <rdfs:domain rdf:resource="#LocalInverse"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<rdf:Property ID="secondDomain">
  <rdfs:domain rdf:resource="#LocalInverse"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Class"/>
</rdf:Property>

<rdf:Property ID="secondRelation">
  <rdfs:domain rdf:resource="#LocalInverse"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<!-- Definitions for COMPOSITION -->

<rdfs:Class ID="Composition"/>

<rdf:Property ID="compose">
  <rdfs:domain rdf:resource="#Composition"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<rdf:Property ID="firstComponent">
  <rdfs:domain rdf:resource="#Composition"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<rdf:Property ID="secondComponent">
  <rdfs:domain rdf:resource="#Composition"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<!-- Definitions for PARTITION -->

<rdfs:Class ID="Partition"/>

<rdf:Property ID="partitionee">
  <rdfs:domain rdf:resource="#Partition"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<rdf:Property ID="parts">
  <rdfs:domain rdf:resource="#Partition"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag"/>
</rdf:Property>

<!-- Definitions for PartonomicRolePropagation -->

<rdfs:Class ID="PartonomicRolePropagation"/>

<rdf:Property ID="propagatedrel">
  <rdfs:domain rdf:resource="#PartonomicRolePropagation"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<rdf:Property ID="propagatingrel">
  <rdfs:domain rdf:resource="#PartonomicRolePropagation"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<rdf:Property ID="topConcept">
  <rdfs:domain rdf:resource="#PartonomicRolePropagation"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Class"/>
</rdf:Property>

<rdf:Property ID="propConcept">
  <rdfs:domain rdf:resource="#PartonomicRolePropagation"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Class"/>
</rdf:Property>

<!-- Definitions for General Axioms-->

<rdfs:Class ID="GeneralAxiom">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#Resource">

```



```
</rdfs:Class>

<rdf:Property ID="lang">
  <rdfs:domain rdf:resource="GeneralAxiom"/>
  <rdfs:range rdf:resource="http://www.w3.org/TR/xmlschema-2/#string"/>
</rdf:Property>

<rdf:Property ID="text">
  <rdfs:domain rdf:resource="GeneralAxiom"/>
  <rdfs:range rdf:resource="http://www.w3.org/TR/xmlschema-2/#string"/>
</rdf:Property>

</rdf:RDF>
```


Knowledge Portals

Ontologies at Work

Steffen Staab and Alexander Maedche

■ Knowledge portals provide views onto domain-specific information on the World Wide Web, thus helping their users find relevant, domain-specific information. The construction of intelligent access and the contribution of information to knowledge portals, however, remained an ad hoc task, requiring extensive manual editing and maintenance by the knowledge portal providers. To diminish these efforts, we use ontologies as a conceptual backbone for providing, accessing, and structuring information in a comprehensive approach for building and maintaining knowledge portals. We present one research study and one commercial case study that show how our approach, called SEAL (semantic portal), is used in practice.

Information on the World Wide Web is ubiquitous, but it is painful to find anything specific. Hence, services flourish that put up knowledge portals for a well-structured orientation on the web.¹ Although there are some general-purpose knowledge portals such as Yahoo, the majority of knowledge portals, however, are domain- or market-specific and serve a particular clientele, for example, Look-Look, which offers structured access to trends in youth culture for companies with an interest in this market.

Knowledge portals typically are maintained manually. Knowledge portal providers enlist hoards of people to contribute information pieces that are shaped by human editors into many different views. The editors' problems include considerations about what information pieces are there, how to structure them, who should look at them, and who should provide them. However, the manual structuring and contribution of large amounts of information for easy access by the users becomes a difficult and expensive problem over time. Therefore, we have developed a method for building and maintaining knowledge portals, the key technology being ontologies to help structure, access, and provide information that has been aggregated by a collaboration of people. For

this purpose, ontologies constitute the formal means that specify the domain of interest for the clientele of the knowledge portal (compare Gruber [1993]).

To date, ontologies have been used for research (Staab et al. 2000; Altman et al. 1999) and commercial purposes such as presenting and mediating information (Wiederhold and Genesereth 1997) on the web,² tackling intriguing parts of the overall problem of building and maintaining knowledge portals. This article presents a comprehensive concept for building and maintaining tasks, including the delivery of decentralized knowledge, as well as tools for accessing, warehousing, and inferring knowledge. We elaborate on the basic tools and methods; a case study serving research needs; and—briefly—a commercial portal currently under development that uses our approach.

Requirements for Knowledge Portals

The aim of knowledge portals is to make knowledge accessible to users and allow users the exchange of knowledge. Knowledge portals specialize in a certain topic to offer deep coverage of the domain of interest and, thus, address a community of users. The portals are commonly built to include community services, such as online forums, mailing lists, and news articles of relevant guises (Faulstich 2000).

Even facing only a medium-size portal, the amount of information that is stored becomes extremely unwieldy to present and refine. In particular, the common categories, such as news or mailings, appear completely inadequate to deal with the information flood on their own. Hence, the question about how best to manage such a knowledge portal becomes urgent. One reason for this is that the user will often not care so much about the document

type (mailing list, magazine article, interviews) but, rather, about the document content when he/she searches for knowledge to solve a problem or learn about a new topic.

In fact, a number of research proposals and commercial solutions exist that have recognized and approached this problem. For example, MATHNET4 introduces knowledge sharing for mathematicians through a database relying on Dublin Core metadata. Altman et al. (1999) allow for navigating their knowledge base on Ribosomes according to an ontology, thus providing rich interlinkage and good support for the user. Further work in this direction in various guises has also been done (for example, Martin and Eklund [1999]; Fernandez et al. [1998]; and Maurer [1996]), but a comprehensive concept for supporting the knowledge portal has been missing thus far. Such overall support has to include, of course, the graphic user interface (GUI) for accessing the portal contents and thereby addressing community-specific needs, but it also needs to consider the contribution of knowledge as well as the overall construction and maintenance of the portal.

Knowledge Providing

An essential feature of a knowledge portal is the easy addition of new information and/or the easy updating of old information in a way such that it can easily be refound. Thus, information can come in many different legacy formats. Nevertheless, presentations of, and queries for, information contents must be allowed in many ways that need to be independent from the way that information was provided originally. The knowledge portal must remain adaptable to the information sources contributed by its providers—not vice versa.

This requirement precludes the application of database-oriented approaches (for example, Maurer [1996]) because they presume that a uniform mode of storage exists that allows for the structuring of information at a particular conceptual level, such as a relational database scheme. In the complex setting of a knowledge portal, one must neither assume that a uniform mode for information storage exists nor that only one particular conceptual level is adequate for structuring information of a particular community. In fact, even more sophisticated approaches such as XML-based techniques that separate content from layout and allow for multiple modes of presentation appear insufficient because their underlying transformation mechanisms (for example, XSLT or XQL [Deutsch et al. 1999; Robie, Lapp, and Schach 1998]) are too inconvenient for integration and presentation of various formats at different conceptual

levels. The reason is that they do not provide the conceptual underpinning required for proper integration of information.

To integrate diverse information, we require another layer besides the common distinction into document, content, and layout, that is, explicit knowledge structures that can structure all the information in different formats for a community at various levels of granularity. Different information formats need to be captured and related to the common ontology: (1) several types of metadata such as available on web pages (for example, HTML metatags), (2) manual provision of data to the knowledge repository, and (3) a range of different wrappers that encapsulate structured and semistructured information sources (for example, databases or HTML documents). The section entitled "Providing Knowledge" addresses these issues in detail.

Knowledge Access

Navigating through a knowledge portal that is unknown is a rather difficult task in general. Information retrieval can facilitate the finding of pieces of text, but its use is not sufficient to provide novice users with the right means for exploring unknown terrain. This navigation turns out to be a problem particularly when the user does not know much about the domain and does not know what terms to search for. In such cases, it is usually more helpful for the user to explore the portal by browsing—given that the portal is well structured and comprehensive. Simple tree-structured portals can be easy to maintain, but the chance is extremely high that an inexperienced user looking for information gets stuck at a dead-end. Therefore, the portal must be able to present a multitude of varying views onto its contents, and different ways should be possible to approach the same content. For example, when looking for an expert in a given, but still vaguely defined domain, either one might query for research papers, or one might search for projects first and then continue to have a glimpse onto corresponding expert home pages.

Here, we must face the trade-off between resources used for structuring the portal (money, manpower) and the extent to which a comprehensive navigation structure can be provided. Because information in the knowledge portal will continually be expanded and updated, a richly interrelated presentation of information usually requires extensive editing, such as is done, for example, for Yahoo. In contrast, knowledge portals should exhibit comprehensive structuring of information virtually for free.

Interesting research, for example, from Fröhlich, Neijdl, and Wolpers (1998) or Kessler

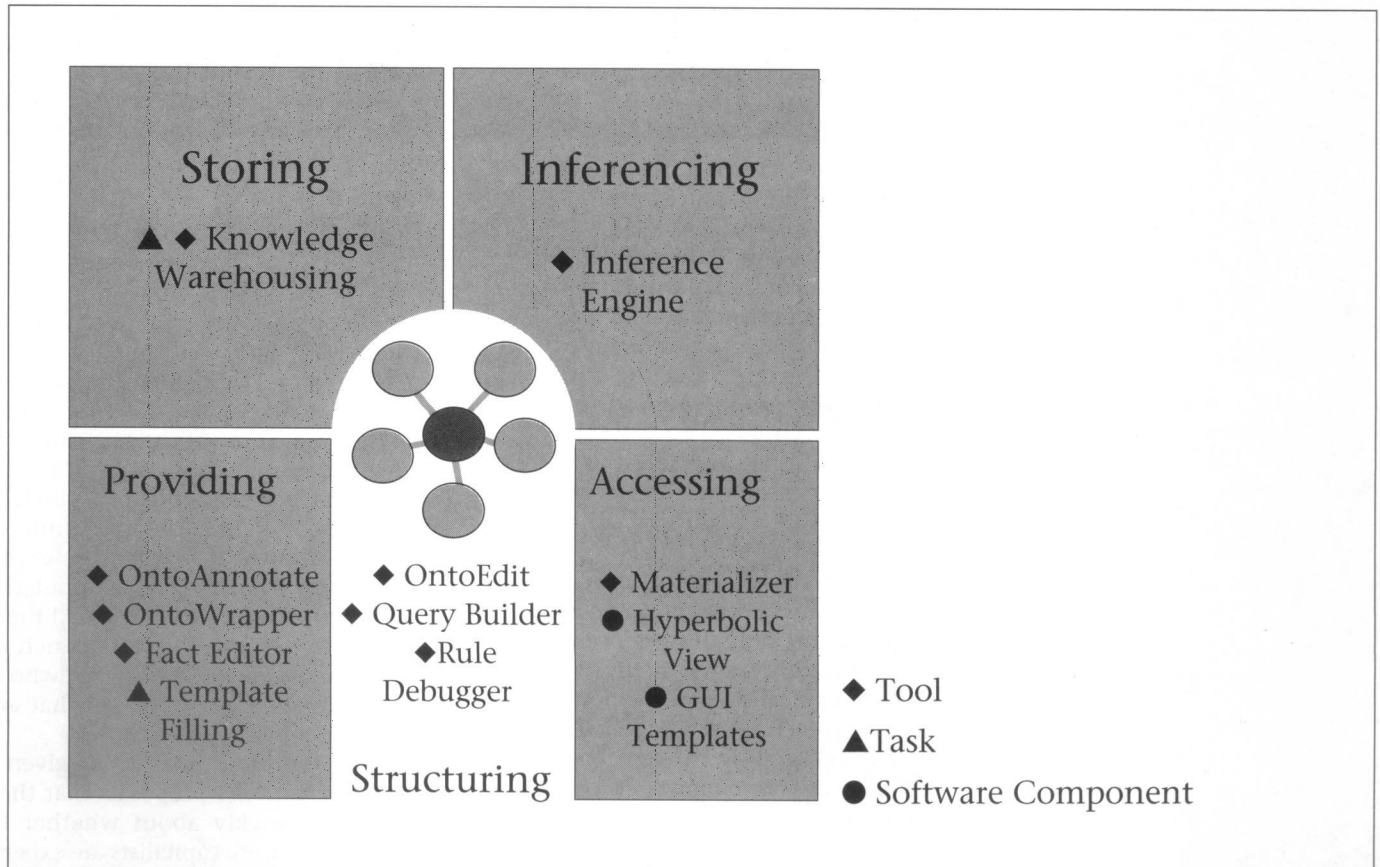


Figure 1. Knowledge Portal Architecture.

(1995), demonstrates that authoring, as well as reading and understanding of web sites, profits from conceptual models underlying document structures *in the large*, that is, the interlinking between documents, as well as document structures *in the small*, that is, the contents of a particular document. In addition, it shows how rich linkage in multiple directions can be constructed automatically based on the underlying conceptual structures.

Naturally, once a common conceptual model for the community exists and is made explicit, it is easier for the individual to access a particular site. Hence, in addition to rich interlinking between document structures in the large, comprehensive surveys and indexes of contents, and a large number of different views onto the contents of the portal, we require that the conceptual structure of the portal be made explicit at some point. We meet this requirement by providing an ontology. The section entitled Access the Knowledge Portal shows how conceptual structures are exploited for access purposes. However, first, we provide a high-level view onto the overall architecture (figure 1).

Architecture

Our architecture is primarily divided into five modules that package different tools, tasks, and software components. We have mentioned some requirements that appear at the interface sides of accessing and providing knowledge, and we elaborate on these in subsequent sections; hence, we can safely ignore them here.

Knowledge Warehousing

The knowledge warehouse hosts facts; metadata about documents; and the ontology, which describes the structure of the facts and the metadata. Facts and concepts are stored in a relational database; however, they are stored in a reified format that treats relations and concepts as first-order objects and, therefore, is flexible with regard to changes and extensions of the ontology.

The different tasks and tools for providing knowledge feed directly into the knowledge warehouse or indirectly when they are triggered by a web crawl. The task at this point is similar to data warehousing, where various schemata need to be mapped to each other, and views need to be maintained and integrated. In the case studies we describe, we could

restrict our attention to incoming data that were already structured according to the given ontology. Thus, the integration task has been rather negligible.

Inferencing

We exploit the inference engine SiLRI (simple logic-based resource description framework [RDF] interpreter) described in Decker et al. (1998). Basically, SiLRI offers representation capabilities for RDF and F-LOGIC and combinations thereof. RDF is a frame-oriented representation language with an XML syntax. F-LOGIC is an object-oriented logic mechanism that extends datalog with object-oriented modeling primitives. Although RDF only allows for the contribution of facts and concept definitions, F-LOGIC also allows the querying and use of axioms.

For our purpose, SiLRI is ideally suited because it allows for the combined querying of facts and ontological concepts. Hence, one can make statements such as "show me the concept taxonomy, including only those concepts for which you have some news in the last week" and, thus, dynamically adapt the portal interface. In our architecture, the knowledge warehouse is only queried by the inference engine, thus offering a uniform mode of access. However, the inference engine caches previous queries to deliver short response times.

Structuring

Finally, we offer several tools for structuring the portal, that is, engineering the ontology that constitutes the background for the inference engine and making contents accessible. We elaborate on these tools in the section "Structuring the Knowledge Portal," but first, we introduce two case studies as our litmus test for the validity of our approach and as illustrations of some examples presented in the remainder of the article.

Case Study: KA2 Portal

The first knowledge portal that we constructed was for the Knowledge Annotation Initiative of the knowledge-acquisition community (KA2) (compare Benjamins, Fensel, and Decker [1999]). The KA2 initiative was conceived for semantic knowledge retrieval from the web, building on knowledge created in the knowledge-acquisition community. To structure knowledge, an ontology has been built by an international collaboration of researchers. The ontology constitutes the basis for annotating web documents from the knowledge-acquisition community to enable intelligent access to these documents and infer implicit knowledge from explicitly stated facts and rules from the ontology.

Given this basic scenario, we have investigated the techniques and built the tools that we describe in the rest of this article. Some views of the KA2 contents can be seen in our up and running demonstration KA2 community web portal (figure 2).³

Case Study: TIME2RESEARCH Portal

TIME is an acronym for telecommunications, information technology, multimedia, and e-business. The term *TIME market* refers to a rapidly evolving market segment with tremendous opportunities. Some of the challenges in this business segment lie in observing the market, tracking (un)successful business models, and evaluating competing or new technologies. In particular, there are many people who are not genuinely knowledgeable about the TIME market and the technologies used there but who need in-depth information such as who is selling what type of technology, who is market leader in subsegment X, or what are peer groups of companies in sector X.

For example, venture capitalists are given a large number of business proposals that they must decide on quickly about whether to invest. Typically, venture capitalists are experts in financial issues of starting a company, and accordingly, they use their financial expertise as a sieve to sort out the good potential investments. In technical matters, they would need some corresponding sieves, which they must commonly buy from a consulting company because having the technical analyst around would be too expensive. From financial and technical points of view, evaluation criteria of different grain sizes are used that take up different amounts of time and money. A successful proposal would run through several evaluation cycles where increasingly fine-grained criteria and increasingly time-consuming evaluation measures are applied to sort out the good potential investments from the bad.

From the outsourcing of technical expertise comes a difficult problem: The duration between proposal and answer is rather long as the evaluation goes through several stages. Thus, investors clog their working line, and more importantly, they can miss good chances because proposers can turn toward other investors. Also, the overall process is not very efficient because many standard questions (such as the ones mentioned at the beginning of this section) must redundantly be researched and answered by different technical experts.

The TIME2RESEARCH knowledge portal aims at streamlining the process that the technical

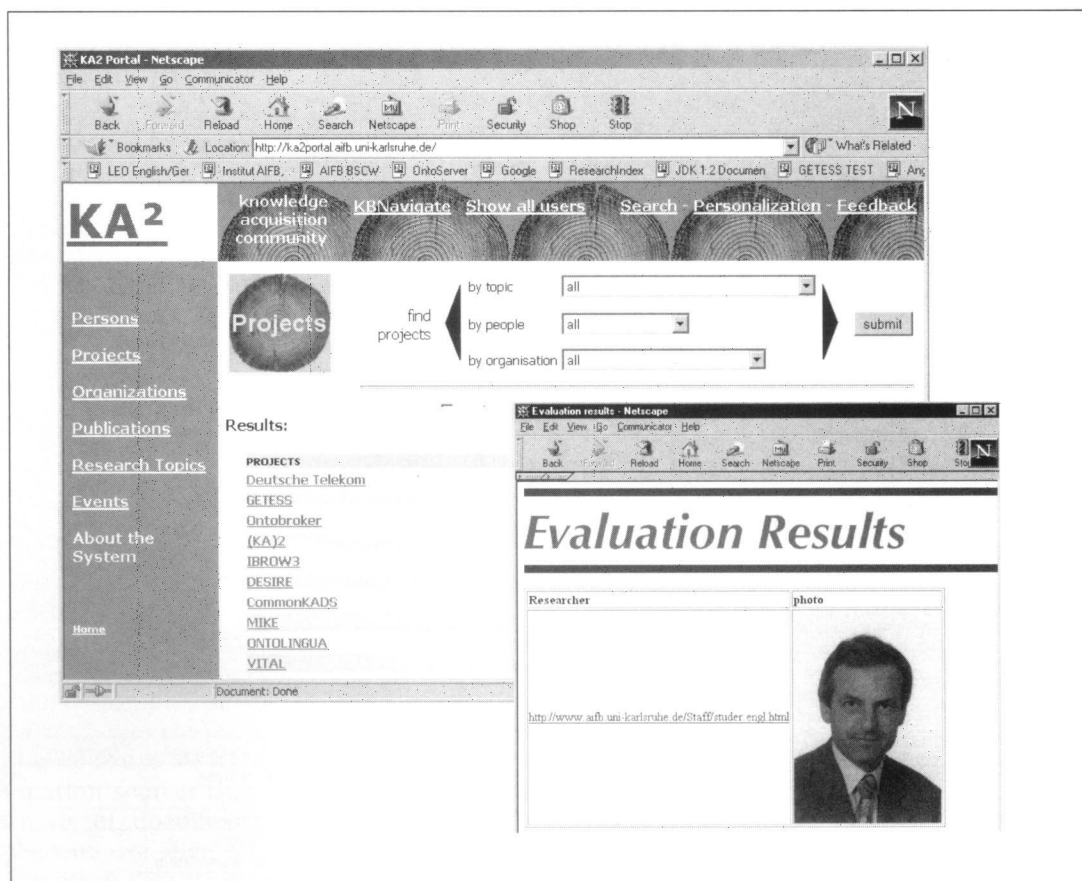


Figure 2. KA2 Portal User Interface.

analyst performs because it allows for collaborative knowledge contribution. The portal optimizes the information-delivery process between the venture capitalist and the technical expert because it allows for decentralized knowledge querying. It allows a bridge between the need for exploring the landscape and the technical expertise because the ontology structures the relevant domain of the TIME market in terms of the one who compiles the question. The TIME2RESEARCH knowledge portal is an intriguing application because ontologies greatly extend the capabilities of current knowledge portals in this area. Thereby, it need not solve the overall problem—evaluation in later stages will still have to be performed by technical analysts—but the venture capitalist can answer his/her standard questions to the portal in a few minutes instead of triggering a day- or week-long process.⁴

Structuring the Knowledge Portal

Ontologies have been established for knowledge sharing and are used as a means for con-

ceptually structuring domains of interest (Wiederhold and Genesereth 1997; Uschold and King 1995). Because knowledge portals focus on particular domains, ontologies appear ideally suited to support knowledge sharing and reuse between knowledge portal providers and the users of the portal. In this section, we describe what representation formats underly the ontologies we use in our knowledge portals and the tools we use for constructing them.

Our domain ontologies consist of (1) concepts defining and structuring domain-specific terms; (2) properties between concepts (that is, relations) and between concepts and built-in types (that is, attributes); and (3) axioms that allow for additional inferences, such as the verification of constraints and the generation of new knowledge.

We model ontologies at an epistemological level using the sophisticated graphic means of the ontology engineering workbench ONTOEDIT (figure 3) (Staab and Maedche 2000).⁵ The workbench offers different views for modeling concepts, attributes, relations, and axioms. The resulting ontology can be translated into different actual representation languages, that is, F-LOGIC, RDF, OIL, and DAML + OIL (table 1).⁶

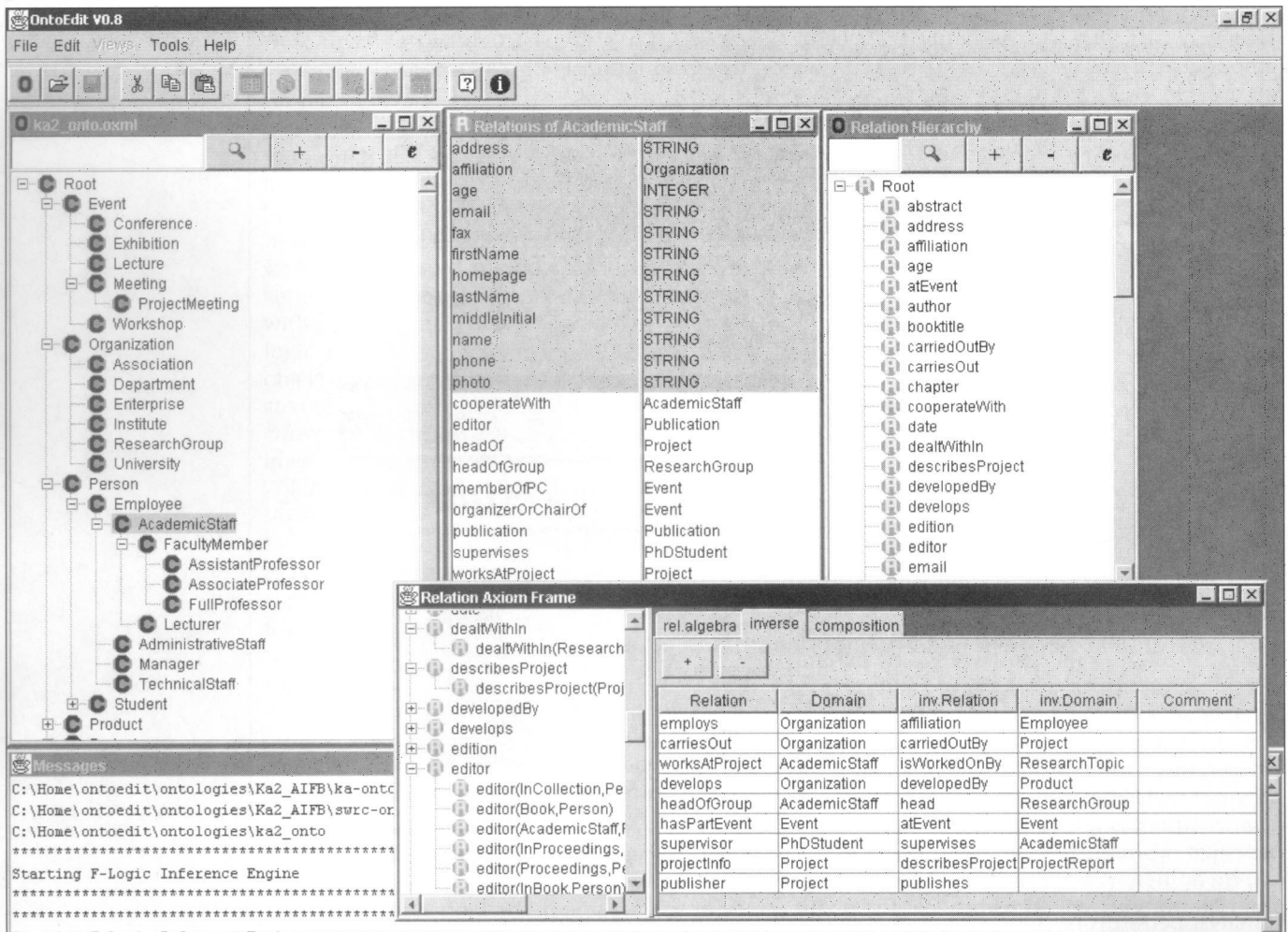


Figure 3. Part of the KA2 Ontology in an ONTOEDIT Screenshot.

To illustrate the structure of the ontologies built with ONTOEDIT, the screenshot in figure 3 depicts part of the KA2 ontology describing a research community as it is seen in the ontology development environment ONTOEDIT. The leftmost window depicts the is-a-relationship that structures the concepts of the domain in a taxonomy. Attributes and relations of concepts are inherited by subconcepts. Multiple inheritance is allowed because a concept might fit into different branches of the taxonomy. In figure 3, attributes and relations of the concept *AcademicStaff* appear in the middle window. Some of these attributes, such as *FirstName* and *LastName*, are inherited from the superordinate concept *Person*. Relations refer to other concepts, such as *WorksAtProject* denoting a relation between *AcademicStaff* and *Project*.

Beyond simple structuring, we model axioms or rules, which are defined on top of the core ontology allowing inferencing and, thus, the generation of new knowledge. For

this purpose, we define semantic patterns (Staab, Erdmann, and Maedche 2001) that describe generic reasoning behavior. One example is the common membership pattern

MembershipRelated(memberrelation, directrelation)

This pattern expresses that if two different instances i_1, i_2 belong to a set S by way of the membership relation *memberrelation*, they are related to each other by the *directrelation*.

Such a generic semantic pattern is instantiated by ontology concepts or relations through the graphic interface. For example, "two persons that belong to a common project are said to collaborate" or "two persons that have written a common paper are coauthors."

MembershipRelated(worksAtProject, cooperatesWith).

MembershipRelated(writesPaper, coauthorOf).

This representation can then be (partially)

translated into different target languages (compare table 1 for a survey), as can be seen in table 2.

Providing Knowledge

"One method fits all" does not meet the requirements we have sketched here for the information contribution part of knowledge portals. What one rather needs is a set of methods and tools that can account for the diversity of information sources of potential interest for presentation at the knowledge portal. Although these methods and tools need to obey different syntactic mechanisms, coherent integration of information is only possible with a conceptual basis that can sort loose pieces of information into a well-defined knowledge warehouse. In our setting, the conceptual basis is given through the ontology that provides the background knowledge and that supports the presentation of information by semantic, that is, rule-enhanced queries. Talking about the syntactic or interface side, we support three major, different modes of information contribution: First, we handle metadata-based information sources that explicitly describe contents of documents on a semantic basis. Second, we align regularities found in documents or data structures with the corresponding semantic background knowledge in wrapper-based approaches. Thus, we can create a common conceptual denominator for previously unrelated pieces of information. Third, we allow the direct contribution and maintenance of facts through our fact editor. In addition to the mechanisms described earlier, we provide the developers of a knowledge portal with an RDF-based crawler that searches the web with ontology focus for relevant instances described as RDF expressions. All the information is brought together in a knowledge warehouse that stores data and metadata alike. Thus, it mediates between the original information sources and the navigating and querying needs discussed in the next section.

Metadata-Based Information

Metadata-based information enriches documents with semantic information by explicitly adding metadata to the information sources. Over the last years, several metadata languages have been proposed that can be used to annotate information sources. In our approach, the specified ontology constitutes the conceptual backbone for the different syntactic mechanisms.

Current web standards for representing metadata such as RDF (Lassila and Swick 1999)

Quite a large number of representation languages for representing ontologies on the web have been established over the last decade. Here, we here give a brief survey of existing ontology representation languages and associated systems on the web:¹

The current starting point for ontology languages on the web are recommendations of the W3C for representing semistructured data on the web with resource description framework (RDF) and for modeling concepts and relations with RDF schema (RDFS).^{2,3} RDF represents the core data model that enables the encoding, exchange, and reuse of semistructured data, comprising a simple triple model for relations together with a convention for expressing reified facts, and also comes with an XML-style syntax. RDFS is an RDF application that basically allows you to describe concept and property hierarchies as well as domain restrictions and range restrictions of properties. RDF and RDFS serve as a lightweight semantic layer that can be mapped onto other languages or that are used as a foundation for other languages.

ONTOBROKER (Decker et al. 1999) and SEAL (semantic portal), our approach for building knowledge portals, use F-LOGIC, an object-oriented and logics-based representation language conceived by Kifer, Lausen, and Wu (1995). It supports inferencing for query answering on schema and instance level, extending horn logic with object-oriented primitives. In the implementation SILRI by Angele and Decker (Decker et al. 1998) that we use, the F-LOGIC engine can integrate RDF and RDFS facts and reason on them.⁴ In a similar category is SHOE (simple HTML ontology extensions) (Heflin and Hendler 2000)⁵, which uses the PAR-KA knowledge representation system, allowing the user to define a frame-based ontology with class, subclass, and property links. Additionally, on top of this frame-based ontology, horn logic rules can be defined.

Conceptual graphs (Sowa 1992) are a system of logics based on the existential graphs of Charles Sanders Peirce and semantic networks. The WEB-KB system (Martin and Eklund 1999) describes an application similar to ONTOBROKER that embeds knowledge in web documents using conceptual graphs. A mapping of RDF into conceptual graphs is described in Corby, Dieng, and Hebert (2000), where a conceptual graph mechanism is used to answer queries about stored RDF facts.

Description logics are a fragment of first-order logic with rather expressive primitives but still decidable and (empirically) efficient inference procedures. LOOM (MacGregor 1991) is a frequently used system with incomplete description logic reasoning that has also been used commercially for web applications. OIL (the ontology inference layer) offers an integration of RDF-RDFS with basically a description logic-based semantics (Decker et al 2000).⁶ Thus, it provides a semantically richer and more precise basis than RDF, embracing the current web standards. There is a mapping of OIL into the efficient terminological reasoning system FACT (Horrocks 1998).

One of the most recent developments in the Defense Advanced Research Projects Agency (DARPA) Agent Markup Language (DAML) Initiative is the proposal of the language DAML-ONT. As a layer on top of RDF/RDFS, DAML-ONT is—like OIL—intended to integrate ontologies with web standards.⁷ Current efforts in DAML aim toward the integration with OIL into DAML + OIL as well as toward the integration of a rule language.

1. Compare van Harmelen and Fensel (1999) for an excellent survey, which naturally cannot anymore cover the current state of affairs completely.
2. W3C. RDF Schema Specification. www.w3.org/TR/PR-rdf-schema/.
3. W3C Recommendation available at www.w3.org/TR/REC-rdf-syntax.
4. www.ontoprise.de/download.
5. www.cs.umd.edu/projects/plus/SHOE/.
6. www.ontoknowledge.org/oil.
7. www.daml.org/2000/10/daml-ont.HTML.

Table1. Overview of Ontology Languages and Systems on the Web.

```

<rdf:rdf
xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:ka2 = "http://www.semanticweb.org/ontologies/ka2-onto-2000-11-07.rdfs#">

  <ka2:FullProfessor
rdf:ID="http://www.aifb.uni-karlsruhe.de/person:rst">
  <ka2:firstName>Rudi</ka2:firstName>
  <ka2:lastName>Studer</ka2:lastName>
  <ka2:homepage
rdf:resource="http://www.aifb.uni-karlsruhe.de/Staff/studer.html"/>
  </ka2:FullProfessor>
</rdf:rdf>

```

Figure 4. RDF Facts Instantiated Using the Vocabulary Given through the KA2 Ontology.

Language	Result	Comment
F-Logic	FORALL x,y,z x[cooperatesWith->>y] <- x[worksAtProject->>z] and y[worksAtProject->>z] and not equal(x,y).	
KIF	(=> (worksAtProject ?x ?z) (worksAtProject ?y ?z) (= ?x ?y) (cooperatesWith ?x ?y))	
SHOE	<DEF-INFERENCE> <INF-IF> <RELATION NAME="worksAtProject"> <ARG POS=1 VAR VALUE="X"/> <ARG POS=2 VAR VALUE="Z"/> </RELATION> <RELATION NAME="worksAtProject"> <ARG POS=1 VAR VALUE="Y"/> <ARG POS=2 VAR VALUE="Z"/> </RELATION> </INF-IF> <INF-THEN> <RELATION NAME="cooperatesWith"> <ARG POS=1 VAR VALUE="X"/> <ARG POS=2 VAR VALUE="Y"/> </RELATION> </INF-THEN> </DEF-INFERENCE>	negation not allowed in SHOE, hence "partial" semantics incur overgener- ation of "cooperatesWith" relationships

Table 2. Resulting Output for Different Ontology Languages.

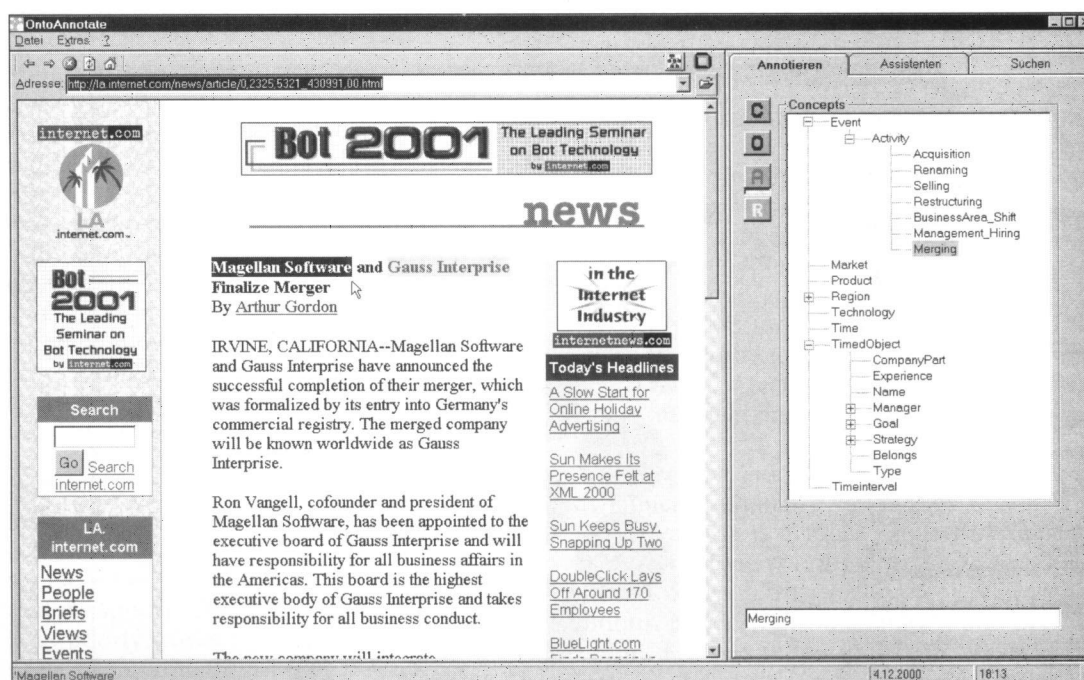


Figure 5. ONTOANNOTATE—Providing Semantics in HTML Documents.

or XML can be handled within our knowledge portal approach.⁷ We have developed a method and a tool called DTDMAKER for generating document-type definitions (DTDs) out of ontologies (Erdmann and Studer 1999). DTDMAKER derives an XML DTD from a given ontology so that XML instances can be linked to an ontology. The link has the advantage of grounding the document structure on a true semantic basis; thus, facts from XML documents can be integrated directly into the knowledge warehouse. The method has the advantage of having the large number of available XML tools, for example, for editing documents, become tools that provide formal metadata for the knowledge portals. HTML-A, early proposed by Fensel et al. (1998), is an HTML extension that adds annotations to HTML documents using an ontology as a metadata schema. HTML-A has the advantage of smoothly integrating semantic annotations into HTML and preventing the duplication of information.

More widespread, RDF facts serve as direct input for the knowledge warehouse, and RDF facts can be generated from information contained in the knowledge warehouse. An example of RDF metadata-based information is given through the following RDF expression, which states that the string *Rudi Studer* is the Name of the instance of the concept *FullProfessor* with the object identifier *www.aifb.uni-karlsruhe.de/person:rst*. Additionally, the home page of the object *www.aifb.uni-karlsruhe.de/person:*

rst is defined by the attribute *Homepage*. These RDF facts are instantiated using the vocabulary given through the KA2 ontology (figure 4).

To facilitate the annotation of HTML, we have developed an RDF-based annotation tool called ONTOANNOTATE (compare figure 5, where a merger between two information technology companies, Gauss and Magellan, is captured). ONTOANNOTATE and its underlying mechanisms for semantic annotation are described in further detail in Erdman et al. (2000). It is also possible to enrich documents generated with Microsoft Office applications with metadata by using our plug-ins WORD-RDF and EXCEL-RDF.

For the future, we envision a semiautomatic tool that combines automatic information-extraction techniques with manual accuracy. We currently do research on this task as part of the DAML ONTOAGENTS Project.⁸

Wrapper-Based Information

In general, annotating information sources by hand is a time-consuming task. Often, however, annotation can be automated when one finds regularities in a larger number of documents. The principle idea behind wrapper-based information is that there are large information collections that have a similar structure. We here distinguish between semi-structured information sources (for example, HTML) and structured information sources (for example, relational databases).

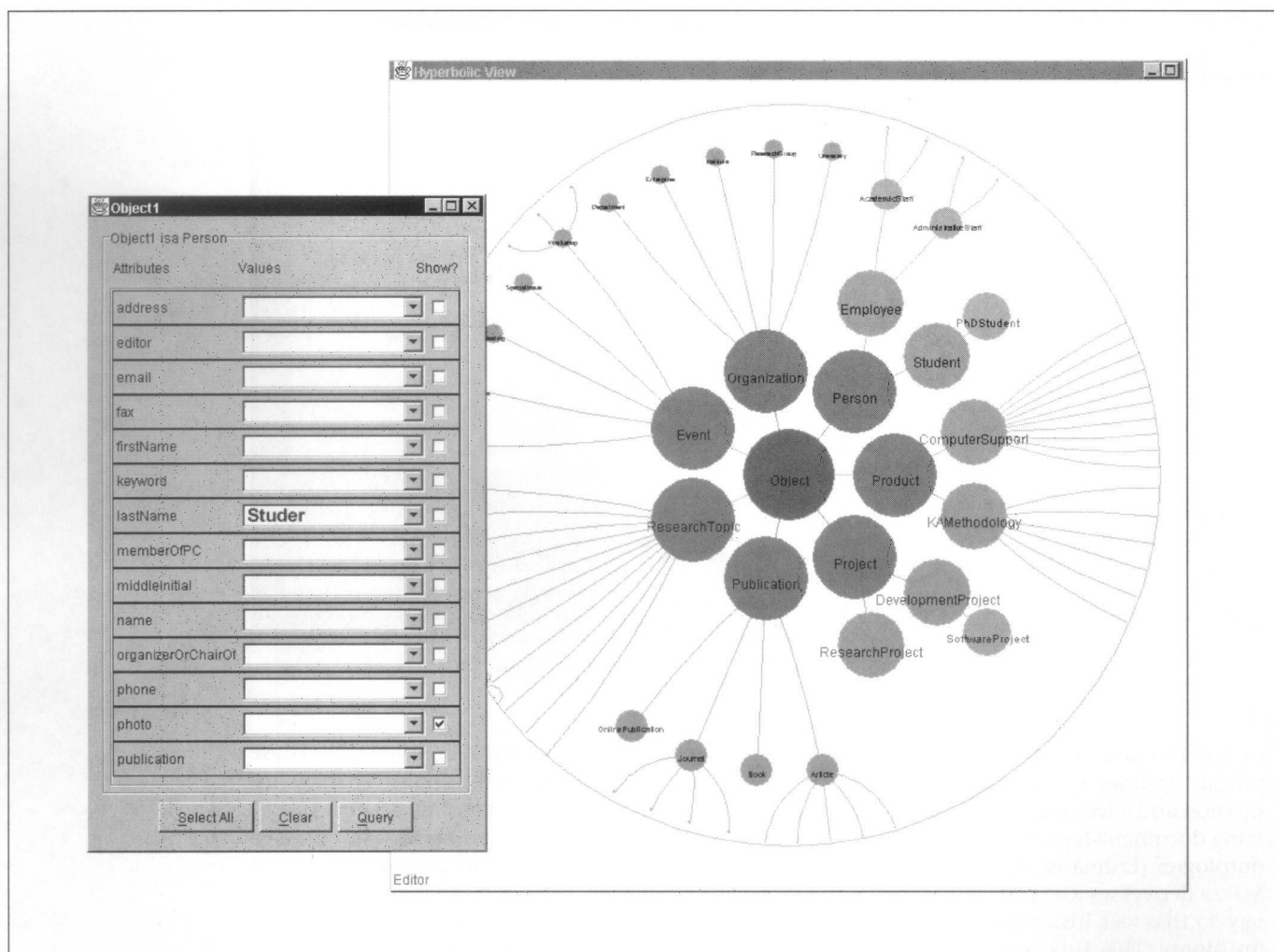


Figure 6. Hyperbolic Query View Interface.

Semistructured Sources

In recent years, several approaches have been proposed for wrapping semistructured documents, such as HTML documents. Wrapper factories (compare Sahuguet and Azavant [2001]) and wrapper induction (compare Kushmerick [2000]) have considerably facilitated the task of wrapper construction. To wrap directly into our knowledge warehouse, we have developed our own wrapper approach that directly aligns regularities in semistructured documents with their corresponding ontological meaning.

Structured Sources

Often, existing databases and other legacy systems can contain valuable information for building a knowledge portal. Ontologies have shown their usefulness in the area of intelligent database integration. They act as information mediators (compare Wiederhold and

Genesereth [1997]) between distributed and heterogeneous information sources and the applications that use these information sources. Existing entities in legacy systems are mapped onto concepts and relations defined in the ontology. Thus, existing information can be pumped into the knowledge warehouse by a batch process, or it can be accessed on the fly.

Fact Editor

The process of providing new facts for the knowledge warehouse should be as easy as possible. For this reason, we offer the *hyperbolic interface tool* (compare figure 6) that can be used as a fact editor. In this mode, its forms are not used to ask for values but to insert values for attributes of instances of corresponding concepts from the ontology. The fact editor is also used for maintaining the portal, that is, to add, modify, or delete facts.

Access the Knowledge Portal

Having provided information with a conceptual underpinning, we now want to provide the same rich semantic structures to define a multitude of views that dynamically arrange information. Thus, our system can yield the kind of rich interlinking that is most adequate for the individual user and his/her navigation and querying of the knowledge portal. We start with a description of the query capabilities in our representation framework. Although in principle, we could use a number of different query languages, in practice, our framework builds on the very same F-LOGIC mechanism for querying as it did for ontology representation; thus, it can also exploit the ontological background knowledge. Through this semantic level, we achieve the independence from the original, syntactically proprietary, information sources that we stipulated earlier. Nevertheless, F-LOGIC is as poorly suited for presentation to naive users as any other query language. Hence, its use is mostly disguised in various easy-to-use mechanisms that more properly serve the needs of the common user, although it still gives the editor all the power of the principal F-LOGIC representation and query capabilities.

Query Capabilities

To illustrate the range of queries used in our portals, we give a few simple examples. For example, using a concrete example from our KA2 portal, the following query asks for all publications of the researcher with the last name Studer:

```
FORALL Pub <- EXISTS ResID
  ResID:Researcher[lastName ->> "Studer";
  publication ->> Pub].
```

The substitutions for the variable Pub constitute the publications queried by this expression.

Besides retrieving explicit information, the query capabilities allow implicit information to be made explicit. They use the background knowledge expressed in the domain ontology, including rules as introduced earlier. If we have a look at web pages about research projects, information about the researchers (for example, their names and their affiliation) involved in the projects is often explicitly stated. However, the fact that researchers who are working together in projects are cooperating is typically left aside. A corresponding question might be, Which researchers are cooperating with other researchers? When querying for cooperating researchers, the implicit information about project cooperation of researchers is exploited. The query can be formulated as

```
FORALL ResID1, ResID2 <- ResID1:
  Researcher[cooperatesWith ->> ResID2]
  and ResID2:Researcher.
```

The result set includes explicit information about a researcher's cooperation relationships, which are stored in the knowledge warehouse, and also implicit information about project cooperation between researchers derived using the project-cooperation rule modeled in the ontology and inferred by SiLRI.

Usually, it is too inconvenient for users to query the portal using F-LOGIC. Therefore, we offer a range of techniques that allow for navigating and querying the knowledge portals we built:

A hypertext link can contain a query that is dynamically evaluated when one clicks on the link. Browsing is made possible through the definition of views onto top-level concepts of the ontology, such as persons, projects, organizations, publications, technology, and organization. Each of these topics can be searched using predefined views. For example, a click on the projects hyperlink results in a query for all projects known at the portal. The query is evaluated, and the results are presented to the user in a table.

A choice of concepts, instances, or combinations of both can be issued to the user in HTML forms. Choice options can be selected through check boxes, selection lists, or radio buttons. For example, entering CHAR, an F-LOGIC query is evaluated, and all existing companies contained in the portal are retrieved and dynamically offered for selecting among activities of companies in a drop-down list. Search or selection can be further restricted using specific attributes contained in the ontology, such as more specific types of activity or shorter time periods.

For the KA2 portal, we have materialized the ontology with all its underlying facts (compare KBNAVIGATE in figure 2). The ontology is offered in a tree view, and a click on a concept directly shows all underlying instances.

A query can also be generated by using the hyperbolic view interface (compare figure 5). The hyperbolic view visualizes the ontology as a hierarchy of concepts. The presentation is based on hyperbolic geometry (compare Lamping, Rao, and Pirolli [1995]), where nodes in the center are depicted with a large circle, whereas nodes at the border of the surrounding circle are only marked with a small circle. This visualization technique allows a survey over all concepts, a quick navigation to nodes far away from the center, and a closer examination of nodes and their vicinity. When a user selects a node from the hyperbolic view, a form is presented that allows the user to select attributes

or insert values for the attributes. An example is shown in figure 5. The user is searching for the community member Studer and his photo. Based on the selected node and the corresponding attributes, a query is compiled. The query-result is shown in the right part of figure 2.

Furthermore, queries created by the hyperbolic view interface can be stored using the personalization feature. Queries are personalized for the different users and are available for the user in a selection list. The stored queries can be considered as semantic bookmarks. By selecting a previously created bookmark, the underlying query is evaluated, and the updated results are presented to the user. Thus, every user can create a personalized view onto the portal (compare personalization in figure 2).

Finally, we offer an expert mode. The most technical (but also most powerful and flexible) way for querying the portal requires that F-LOGIC be typed in by the user. This way is only appropriate for users who are very familiar with F-LOGIC and the domain ontology.

Conclusion

Knowledge portals serve as intermediaries for knowledge access and knowledge sharing on the web. We have demonstrated how ontologies can lay a conceptual foundation that supports the building of knowledge portals, including means for knowledge access and contribution. The two case studies that we showed appear only as the tip of the iceberg of applications yet to come. Already now, the first electronic-commerce portals have started embracing ontologies, and corporate portals for managing enterprise internal knowledge are catching up (Staab et al. 2001). Nevertheless, full-fledged support of ontology-based technology on the web has been missing until now, and our approach needs to be extended in many directions, such as additional means for ontology-based personalization or log mining with conceptual structures.

We think that our work on knowledge portals is only one very early starting point toward the semantic web that will provide machine-readable infor-

mation for all kinds of web-based applications. In particular, future applications will need to integrate more automatic techniques—for building ontologies (Maedche and Staab 2000), providing metadata, and learning from the use of the semantic web.

Acknowledgments

The research presented in this article would not have been possible without our colleagues and students at the Institute for Applied Informatics and Formal Description Methods, University of Karlsruhe, and Ontoprise GmbH. We thank Jürgen Angele, Kalvis Apsitis (now with RITI Riga Information Technology Institute), Stefan Decker (now with Stanford University), Michael Erdmann, Dieter Fensel (now with VU Amsterdam), Siegfried Handschuh, Andreas Hotho, Mika Maier-Collin, Daniel Oberle, Hans-Peter Schnurr, Rudi Studer, and York Sure. Research for this article was partially financed by Ontoprise GmbH, Karlsruhe, Germany; the U.S. Air Force as part of the Defense Advanced Research Projects Agency DAML OntoAgents Project; European Union as part of the IST-1999-10132 On-To-Knowledge Project; and BMBF as part of the GETESS Project (01IN901C0).

Notes

1. B. Templin, 1999, Dethroning the Content King, available at webreview.com/pub/1999/06/25/feature/content.HTML.
2. Chemdex—A Vortex Business. www.ontology.org/main/papers/cases/chemdex.HTML.
3. ka2portal.aifb.uni-karlsruhe.de.
4. A demo version of the portal is set up during the printing of this article at www.time2research.de/.
5. A simplified version of ONTOEDIT is available for download at www.ontoprise.com.
6. Compare ontobroker.semanticweb.org/ontos/ for a number of ontologies in F-LOGIC, OIL, and DAML-ONT; also compare table 1 on different ontology languages for the web.
7. W3C. XML Specification. www.w3.org/XML/.
8. WWW-DB.Stanford.EDU/OntoAgents/.

References

- Altman, R. B.; Bada, M.; Chai, X. J.; Whirl Carillo, M.; Chen, R. O.; and Abernethy, N. F. 1999. RIBOWEB: An Ontology-Based System for Collaborative Molecular Biology. *IEEE Intelligent Systems* 14(5): 68–76.
- Angele, J.; Schnurr, H.-P.; Staab, S.; and Studer, R. 2000. The Times They Are A-Changin'—The Corporate History Analyzer. Paper presented at the Third International Conference on Practical Aspects of Knowledge Management, 30–31 October, Basel, Switzerland.
- Benjamins, R.; Fensel, D.; and Decker, S. 1999. KA2: Building Ontologies for the Internet: A Midterm Report. *International Journal of Human-Computer Studies* 51(3): 687.
- Corby, O.; Dieng, R.; and Hebert, C. 2000. A Conceptual Graph Model for W3C Resource Description Framework. In *Proceedings of the ICCS 2000—International Conference on Conceptual Structures*. Lecture Notes in Artificial Intelligence, 468–482. Berlin: Springer-Verlag.
- Decker, S.; Brickley, D.; Saarela, J.; and Angele, J. 1998. A Query and Inference Service for RDF. Paper presented at the W3C Query Language Workshop (QL-98), 3–4 December, Boston, Massachusetts.
- Decker, S.; Erdmann, M.; Fensel, D.; and Studer, R. 1999. ONTOBROKER: Ontology-Based Access to Distributed and Semi-Structured Information. In *Database Semantics: Semantic Issues in Multimedia Systems*, eds. R. Meersman, Z. Tari, and S. M. Stevens, 351–369. New York: Kluwer Academic.
- Decker, S.; Fensel, D.; van Harmelen, F.; Horrocks, I.; Melnik, S.; Klein, M.; and Broekstra, J. 2000. Knowledge Representation on the Web. Paper presented at the 2000 International Workshop on Description Logics (DL2000), 17–19 August, Aachen, Germany.
- Deutsch, A.; Fernandez, M.; Florescu, D.; Levy, A.; and Suciu, D. 1999. A Query Language for XML. In *Proceedings of the Eighth International World Wide Web Conference (WWW'98)*, 1155–1169. New York: Elsevier.
- Erdmann, M., and Studer, R. 1999. Ontologies as Conceptual Models for XML Documents. Paper presented at the Twelfth International Workshop on Knowledge Acquisition, Modeling, and Management (KAW'99), 16–21 October, Banff, Canada.
- Erdmann, M.; Maedche, A.; Schnurr, H.-P.; and Staab, S. 2000. From Manual to Semiautomatic Semantic Annotation: About Ontology-Based Text Annotation Tools. Paper presented at the COLING 2000 Workshop on Semantic Annotation and Intelligent Content, 5–6 August, Luxembourg.

- Faulstich, R. 2000. Internet Portals for Electronic Commerce. Master's thesis, Institute for Applied Informatics and Formal Description Methods, University of Karlsruhe.
- Fensel, D.; Decker, S.; Erdmann, M.; and Studer, R. 1998. ONTOBROKER: The Very High Idea. Paper presented at the Eleventh International Flairs Conference (FLAIRS-98), 1-5 May, Sanibel Island, Florida.
- Fernandez, M.; Florescu, D.; Kang, J.; and Levy, A. 1998. Catching the Boat with STRUDEL: Experiences with a Web-Site Management System. In Proceedings of the 1998 ACM International Conference on Management of Data (SIGMOD'98), 414-425. New York: Association of Computing Machinery.
- Fröhlich, P.; Neijdl, W.; and Wolpers, M. 1998. KBS-HYPERBOOK—An Open-Hyperbook System for Education. Paper presented at the Tenth World Conference on Educational Media and Hypermedia (EDMEDIA'98), 20-21 June, Freiburg, Germany.
- Gruber, T. R. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 6(2): 199-221.
- Heflin, J., and Hendler, J. 2000. Dynamic Ontologies on the Web. In Proceedings of the Seventeenth National Conference on Artificial Intelligence, 443-449. Menlo Park, Calif.: American Association for Artificial Intelligence.
- Horrocks, I. 1998. Using an Expressive Description Logic: Fact or Fiction? In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, 636-649. San Francisco, Calif.: Morgan Kaufmann.
- Kessler, M. 1995. A Schema-Based Approach to HTML Authoring. Paper presented at the Fourth International World Wide Web Conference (WWW'4), 11-14 December, Boston, Massachusetts.
- Kifer, M.; Lausen, G.; and Wu, J. 1995. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM* 42(4): 741-843.
- Kushmerick, N. 2000. Wrapper Induction: Efficiency and Expressiveness. *Artificial Intelligence* 118(1): 15-68.
- Lamping, L.; Rao, R.; and Pirolli, P. 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, 401-408. New York: Association of Computing Machinery.
- Lassila, O., and Swick, R. 1999. Resource Description Framework (RDF). Model and Syntax Specification. Technical Report, W3C, Boston, Massachusetts.
- MacGregor, R. 1991. Inside the LOOM Description Classifier. *SIGART Bulletin* 2(3): 88-92.
- Maedche, A., and Staab, S. 2001. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems* (Special Issue on Semantic Web) 16(2).
- Martin, P., and Eklund, P. 1999. Embedding Knowledge in Web Documents. In *Proceedings of the Eighth International World Wide Web Conference (WWW'8)*, 1403-1419. New York: Elsevier Science.
- Maurer, H. 1996. *Hyperwave: The Next Generation Web Solution*. Reading, Mass.: Addison-Wesley.
- Robie, J.; Lapp, J.; and Schach, D. 1998. XML Query Language (XQL). Paper presented at the W3C Query Language Workshop (QL-98), 3-4 December, Boston, Massachusetts.
- Sahuguet, and Azavant, F. 2001. Building Intelligent Web Applications Using Lightweight Wrappers. *Data and Knowledge Engineering* (Special Issue on Intelligent Information Integration) 36(3): 283-316.
- Sowa, J. F. 1992. *Conceptual Structures: Information Processing in Mind and Machine*. Reading, Mass.: Addison-Wesley.
- Staab, S., and Maedche, A. 2000. Ontology Engineering beyond the Modeling of Concepts and Relations. Paper presented at the ECAI-2000 Workshop on Ontologies and Problem-Solving Methods, 21-22 August, Berlin, Germany.
- Staab, S.; Erdmann, M.; and Maedche, A. 2001. Engineering Ontologies Using Semantic Patterns. Paper presented at the IJCAI-2001 Workshop on E-Business and the Intelligent Web, 5 August, Seattle, Washington.
- Staab, S.; Schnurr, H.-P.; Studer, R.; and Sure, Y. 2001. Knowledge Processes and Ontologies. *IEEE Intelligent Systems* (Special Issue on Knowledge Management) 16(1): 26-35.
- Staab, S.; Angele, J.; Decker, S.; Erdmann, M.; Hotho, A.; Maedche, A.; Schnurr, H.-P.; Studer, R.; and Sure, Y. 2000. Semantic Community Web Portals. In *WWW9/Computer Networks* (Special Issue: WWW9—Proceedings of the Ninth International World Wide Web Conference, Amsterdam, The Netherlands, May, 15-19, 2000) 33(1-6): 473-491.
- Uschold, M., and King, M. 1995. Toward a Methodology for Building Ontologies. Paper presented at the Workshop on Basic Ontological Issues in Knowledge Sharing, Fourteenth International Joint Conference on Artificial Intelligence, 20-25 August, Montreal, Canada.
- van Harmelen, F., and Fensel, D. 1999. Practical Knowledge Representation for the Web. Paper presented at the Workshop on Intelligent Information Integration (III99), Sixteenth International Joint Conference on Artificial Intelligence, 31 July-6 August, Stockholm, Sweden.
- Wiederhold, G., and Genesereth, M. 1997. The Conceptual Basis for Mediation Services. *IEEE Intelligent Systems* 12(5): 38-47.



Steffen Staab received an M.S.E. from the University of Pennsylvania in 1994 and a Dr. rer.nat. from the University of Freiburg in 1998, both in informatics. After consulting with the Fraunhofer Institute for Industrial Engineering, Stuttgart, he joined the University of Karlsruhe, where he is now an assistant professor. In 1999, he cofounded Ontoprise GmbH, a company providing a wide range of technologies centering on ontologies. Staab has been working and publishing in computational linguistics, text mining, knowledge management, ontologies, and the semantic web. He won Best Paper Award for a paper on constraint reasoning at ECAI-1998, and he is chairing the Semantic Web Workshop in Hong Kong at WWW 10. His e-mail address is sst@aifb.uni-karlsruhe.de.



Alexander Maedche is a Ph.D. student at the Institute of Applied Informatics and Formal Description Methods, University of Karlsruhe. In 1999, he received a diploma in industrial engineering, majoring in computer science and operations research, also from the University of Karlsruhe. His diploma thesis on knowledge discovery earned him a Best Thesis Award at the University of Karlsruhe. Maedche's research interests cover knowledge discovery in data and text, ontology engineering, learning and application of ontologies, and the semantic web. Recently, he started to build a new research group at the FZI Research Center for Information Technologies at the University of Karlsruhe that researches semantic web technologies and applies them to knowledge management applications in practice. His e-mail address is ama@aifb.uni-karlsruhe.de.

Smart task support through proactive access to organizational memory

S. Staab^{a,*}, H.-P. Schnurr^b

^aInstitute AIFB, University of Karlsruhe¹, 76128 Karlsruhe, Germany

^bOntoprise GmbH, Haid-und-Neu-Straße 7, D-76131 Karlsruhe, Germany

Received 11 July 2000; accepted 31 July 2000

Abstract

We describe an approach towards integrating the semantics of semi-structured documents with task-support for (weakly structured) business processes and proactive inferencing capabilities of a desk support agent. The mechanism of our *Proactive Inferencing Agent* is motivated by the requirements posed in (weakly structured) business processes performed by a typical knowledge worker and by experiences we have made from a first trial with a *Reactive Agent Support* scheme.

Our reactive scheme is an innovative approach for smart task support that links knowledge from an organizational memory to business tasks. The scheme is extended to include proactive inferencing capabilities in order to improve user-friendliness and to facilitate modeling of actual agent support. In particular, the improved scheme copes with varying precision of knowledge found in the organizational memory and it reasons proactively about what might be interesting to you and what might be due in your next step. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Smart task support; Proactive access; Organizational memory

1. Introduction

Intelligent information agents are often compared against their human counterparts, such as secretaries or other colleagues. The comparisons usually find that human assistants

- observe what you do;
- think about what you do;
- have expectations about what you do next;
- thus cope with incomplete information, and still if you ask them they;
- respond quickly to your questions.

Though all of these properties are highly desirable for your personal information agent, so that they answer you fast and save tedious working schemes, nowadays agents lack most of these properties. We present here an architecture and techniques that (partially) implement these properties within a realistic setting, providing information agents that

- reason proactively about what you might be doing;

- take into account what information you and your colleagues have provided so far;
- build up reasonable models about what information you might deserve next; thus
- narrow down choices for potential answers, even if information is missing, and; hence
- respond to you much faster and much more precisely.

Subsequently, we will first describe our motivation for building proactive information agents and describe the *Scenario* into which they are integrated. Then, we describe our starting point, viz. *Reactive Agent Support*, which works on knowledge from an organizational memory created through the usual work tasks of the knowledge worker and which offers queries to the user that may be relevant for her current business task. Based on experiences we have collected with the reactive approach, we have conceived our enhanced agent that adapts more flexibly to the current business task — depending on the amount of knowledge that is actually available for solving this task. This *Proactive Inferencing Agent* builds on the same basic modules as the reactive one, but it employs a refined inferencing strategy that allows for earlier, faster and still more frequent support than the reactive approach. Thereby, it reduces the burden on the human who models the information agent for a particular application and balances automatically between the

* Corresponding author. Tel.: +721-608-4751; fax: +721-693717.

E-mail addresses: staab@aifb.uni-karlsruhe.de (S. Staab), schnurr@ontoprise.de (H.-P. Schnurr).

¹ <http://www.ontoprise.de>.

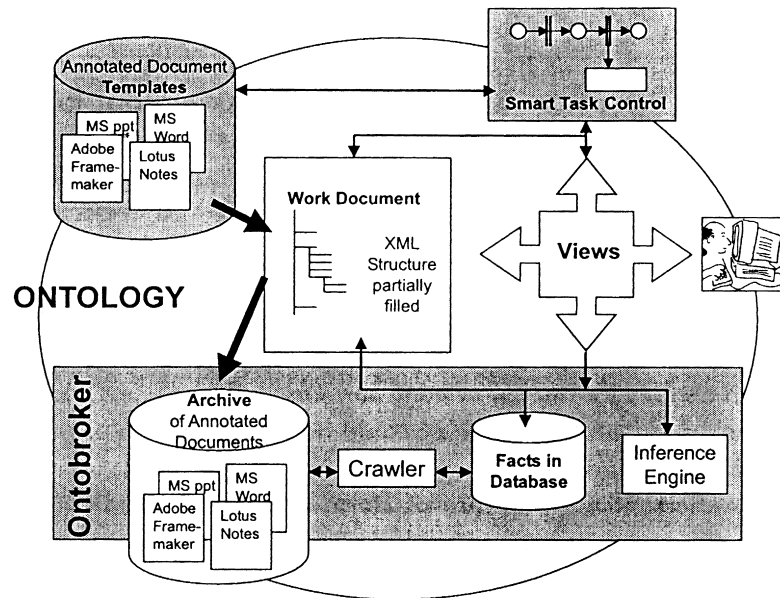


Fig. 1. Reactive agent-support framework.

need for specific answers and the requirement to produce an helpful answer for a broad variety of possible input data. Some examples will elucidate what we pursue with and how we achieve this modeling balance.

2. Scenario

Our approach is embedded in a typical knowledge management setting. In such work desk settings, proactive information agents have often been devised as useful helpers that may facilitate the task the knowledge worker tries to complete on her computer. For instance, project management involves the compilation of a project plan, allocating resources and people in an appropriate way. The project manager must compile a team from within a large company where she is hardly able to know everyone with his capabilities and experiences. Thereby, she must meet the following planning requirements:

- participants must be available for the project;
- participants should have particular technical knowledge that is needed for the project; and
- they should have some knowledge about the client in this project or at least about a client with a similar industry background.

Currently, there are several approaches to handle this problem. First, the team might be compiled from a set of people the manager knows by chance. Second, project listings might circle in the company. Third, all the information might be maintained by a human resource department, e.g.

in a central database. However, all these possibilities come with personnel overhead, time lags, or a lack of quality.

Hence, for our scenario we have the following — realistic — assumptions. In the first assumption, the project manager compiles a project plan that she uses to estimate the man power and expertise she needs for the project, e.g. with a project planning software that supports the creation of network plans, or just with common spreadsheet or text processing software. What is important at this point is that persons who execute this task on a regular basis usually hold on to a particular tool and a particular way of executing this task with this tool. For instance, it is rather typical that a project manager creates a template (or uses a template that is provided by her company) in order to execute and document the planning task. In the second assumption, the information that she relies on is drawn from her personal knowledge, from the knowledge of people she asks, and from the knowledge available in other project documents and in the intranet. Naturally, only the third type of knowledge digitally available in the organizational memory is the one that can be accessed electronically and, thus, it is the one we want to exploit for our knowledge support mechanisms. A common example for a document from the organizational memory could be a project web page describing the name, the goal, the participants and techniques used in the project.

Let us now assume that a groupware platform exists that handles scheduling tasks. The knowledge that is necessary in order to fulfill the three requirements mentioned at the beginning of this section may be found as follows. Availabilities may be retrieved from the scheduling database, and technical knowledge may be inferred from employees' participations in projects at project web pages. Obviously, it is very tedious, sometimes nearly impossible for the

Table 1
The XML structure for a project plan

< project >	
< author > </author >	
< plandate > </plandate >	
< participants >	
</participants >	
< Ganttchart > </Ganttchart >	
< tasks >	
</tasks >	
</project >	

< member > </member >	
< task > </task >	

project manager to gather the information she needs from these different sources. In the next section, we will show how the knowledge for the project planning task may be provided through Reactive Agent Support, once the project planning task has been analyzed and an appropriate methodology and IT support has been introduced to the enterprise.

The objective for our Proactive Inferencing Agent, sketched thereafter, is more akin to having a competent person around you who tutors your project planning. For this purpose, the agent is supposed to reason about what you might ask him next, while you keep on typing (or speaking) into your computer. For instance, you start your project plan with an abstract, noting title, client and time interval. Then you might want to search for people who might participate in the project. Thus, you might pose a question like “Who has experience with XML?” to the agent and you expect an answer almost before you actually pose your question — because Joe Doe is the only one who has experience with XML at all and you noted at the very beginning that you require this type of knowledge. In addition to improved user-friendliness, the proactive approach solves several difficulties that we have experienced while modeling agent support in the reactive framework. The proactive scheme takes considerable burden off the human modeler as it adapts flexibly to different granularities of factual knowledge in the organizational memory.

3. Reactive agent support

In order to provide a concise picture of our approach, we give an overview of the main modules (cf. Fig. 1) and capabilities, before we integrate these different building blocks into our Reactive Agent Support Framework. A detailed description of the main modules is given in Ref. [19].

3.1. Ontobroker for accessing the organizational memory

For our organizational memory infrastructure we employ common intranet technology augmented by semantic access capabilities. A typical intranet environment comprises at least two principal technical services that we rely on. First, an intranet environment offer means to store docu-

ments. In particular, current technology tends to make the boundary between file servers and intra-/internet servers, on the one hand, and web pages and files, on the other hand, disappear. Hence, we may assume that all documents are available in the intranet. Second, intranet environments offer technology for navigating the intranet environment and finding information by common means such as browsing web pages.

Regarding semantic access capabilities, our KM methodology builds on the framework given through the Ontobroker approach as described in Refs. [3,6,18]. The main components of Ontobroker from a functional view point are: (i) the underlying ontology that defines concepts, attributes, relations and rules about a domain; (ii) the annotated document sources, providing facts structured by annotations related to the underlying ontology; (iii) a crawler that gathers facts from the documents and stores them into a database; and (iv) the inference and query engine that reasons on the database to derive facts and that delivers answers to the user who queries it.

3.2. Annotated document templates

Current business documents come in many different guises, such as letters, faxes, order forms, notifications, receipts, memos, private home pages, or project home pages. Nevertheless, it is quite common that these different forms are not chosen arbitrarily, but rather these forms are often standardized up to a certain point, indeed they often come with a particular semantics, such as the short notes that allow the reader to determine sender, reader, urgency, and further actions that need to be taken with a very short glimpse (e.g. check boxes for “please answer”). Similarly, letters are usually not allowed to come in a completely free form, but they are composed with a particular corporate identity in mind. This corporate identity defines fonts, but it also pervades the way a company presents itself to the outside world.

With our approach we go even one step further, since we also link these documents with corporate identity styles to the enterprise’s ontology (or ontologies). This leads us to annotated document templates, and, thus, makes the contents of common business documents available for a semantically structured organizational memory.

Standard Generalized Markup Language (SGML) and a subset of it, XML (eXtensible Markup Language; [20]), are standardization efforts that aim at a general scheme for exchanging documents and document contents. Given the widespread support among major computer software providers that XML has found recently, it is reasonable to assume that the structure of any business document will be accessible by way of XML annotation and query tool in the very near future.

In our scenario, this is also of particular interest, because SGML/XML gives us the power to reason about document structures and contents. For instance, the XML-tags in the pseudo document from Table 1, i.e. a document without

Table 2
Filled template

```

< project >
< author > Jill
Dole < /author >

      < plandate > October 18th, 1999 < /
plandate >

< participants >
      < member > Jill Dole < /member >
      < member > Hans-Peter Schnurr < /
member >
      < member > Steffen Staab < /member >

< /participants >
< Ganttchart > here
goes the table < /
Ganttchart >
< tasks >
      < tasks > Analysis of Nordic Life Business
Processes < /tasks >
      < tasks > Analysis of Nordic Life IT
environment < /tasks >

< /tasks >
< /project >

```

actual content, might serve as a template for project descriptions in general. The (XML) annotations describe the semantics (and, possibly, some layout) of the document structures. When the user fills in parts of the document (Table 2) in order to complete her business task, then she connects the information she provides with corresponding metainformation. Thereby, XML structures may either be derived directly from the ontology [7], or they may be manually specified and mapped onto the ontology.

3.3. Business processes

The controlled interaction between the contribution to document contents and the performance of business tasks is of particular concern to an information agent that aims at the delivery of relevant information at the right time. Hence, we build on a special version of Petri nets, viz. so-called SGML nets by Weitz [22], that allow to formulate the progress of the business process in terms of the document contents. Weitz's approach mostly aims at mechanisms that

cover the comparatively rigid parts of the workflow, therefore his approach is not adequate to model all the — typically unordered — actions of a knowledge worker. Nevertheless, the scheme is most suitable to distinguish between different high-level, well-ordered business tasks, and, thus, to capture their specific *business contexts* and the views that are relevant during their execution by the worker (also cf. Ref. [19]).

For example, the high-level goal of project planning may be separated into several, distinct, well-ordered tasks (Fig. 2). Each of the tasks requires particular knowledge — and, hence, specific help from the intelligent assistant. The SGML-net mechanism allows to capture prerequisites for changing from one task to the other and to define task-specific views onto the organizational memory. Thereby, the current work document(s) reflect the task(s) that must be performed next, e.g. a project team needs to be compiled. This work involves communicating with prospective project participants. The problem often lies in identifying appropriate participants and in establishing the communication link. As for the first, our methodology allows the establishment of blueprint questions like “Which person in the company knows about X and has capacity for projects as of Y?” As for the second, given that a person is identified, the fax number(s) or e-mail addresses may be retrieved simply by giving the name information. Thereby, it is not required that the fax number is stated in a corresponding database entry. Rather it may only be given in the signature of a mail in the general accessible mail archives, or on a home page or only indirectly: via the group that this person belongs to and a rule that states an implication. Hence, through context-specific questions and semantic inferencing the intelligent assistant supports the manager in planning her project.

To integrate the context-based views and SGML nets, we annotate transitions with logic predicates that define when a transition may be executed. In addition, we define logic predicates denoting views onto the organizational memory at transitions and at places and, thus, augment the views with context-specific criteria about their applicability. Hence, we allow to describe what major steps may follow subsequently and what views may be required to enable the completion of a particular step.

3.4. Example for reactive processing

Considering a concrete example aligned to our scenario, our tool has to support a project manager who compiles a team to implement some Knowledge Management Tool at an insurance company in May 2010. The first step for her is to find people in the company who know about the clients ERP Tool A, and have experience with XML, are available at that date, ideally have experience in the insurance industry and want to participate at that project. To ask the prospective team members about their interest in her project, she decides to send a fax. Therefore she starts the word processing software, opens the fax template “Contacting prospective team

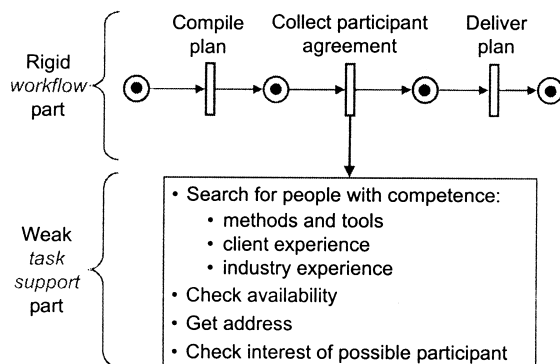


Fig. 2. Integrating workflow and context-based views.

Smart Task Support Datei Bearbeiten Ansicht Einfügen Format Extras Tabelle Fenster ?

Ontobroker

OntoViews

- get people with knowhow "XML"
- get people with availability "May - June 2000"
- get people with client experience "insurance"
- get people with knowhow "XML" and availability "May-June 2000"
- get people with knowhow "XML" and client experience "insurance"
- get people with availability "May - June 2000" and client experience "insurance"
- get people with knowhow "XML", availability "May-June 2000" and client experience "insurance"

Project Plan: Jihaa

Project Management: Jill Dole

Last Update: October 18

State: in preparation

Client: Nordic Life

Industry:

General Goal: Bringing Knowledge Management to Nordic Life

Project Schedule: May - June 2000

Team Requirements: XML

Dear _____,

We are currently discussing a project with one of our key customers

5 1 Ab 1 1/1 Bei 2,4 cm Ze 1 Sp 1 MAK AND ERW UB

Fig. 3. Fax template "Contacting prospective team members".

members" and fills in the right column of the template form (Fig. 3).

The template serves as an input interface for the inference and query engine of the Ontobroker system. Several queries are modeled that depend on which template fields are filled (e.g. cf. left column in Fig. 3). A click to the menu bar yields a list of possible queries and a click to the corresponding query starts the inference engine. In our example, the project manager selects the query 'get people with knowhow "XML"'. The inference engine combines the available database information (gathered from annotated project web pages, project plans and home pages of employees; cf. Tables 3 and 4) and derives the answer: XML was used in

the project "Xfiles". A rule in the ontology concludes that all the participants of a project have experience with methods of that project. The person who fits the description ("Joe Doe") is placed in the header of the fax cover page.

The example also shows the drawbacks of the reactive agent support. Fortunately, in this example, the user finds exactly one answer. This would not be the case, if a user asked for people with knowhow about ERP Tool A. This latter search would require more detailed modeling of a refined query. However, applying a more specific query to the task at hand causes new problems in general, e.g. with the first example, since the corresponding, more specific, query for "XML" and availability from May-June'

Table 3
Knowledge about projects

	Project	Participant	Purpose	Methods	Client
1	"Xfiles"	Joe Doe	Knowledge management support with Ontobroker	XML, Ontobroker	Southern insurance
2	"Yawn"	Fred Bloggs, Walter Moe	Yet another accounting scheme	ERP Tool A	Kwik Fit
...
8	"Guru"	Jane Smith	Great user utility through nothing	ERP Tool A	WonderTool

Table 4
Knowledge about employees

Employee	Telephone	Fax	eMail	Availability in 2010
Fred Bloggs	234567	991234	fred@onto.de	July–October
Joe Doe	345678	992345	joe@onto.de	August–December
Walter Moe	987654	993456	walter@onto.de	July–November
Jane Smith	456789	994667	jane@onto.de	April–November
...	

would not result in any answer at all. The careful balance between specificity and generality of queries is currently achieved by the user who selects from a list of queries. As one may easily see in Fig. 3, this strategy puts too much burden on the user since she must choose from a long menu list of difficult and steadily changing queries. A more suitable and user friendly support would be to give hints about possible answers in a proactive manner. In addition, the reactive agent scheme requires time consuming modeling of a large number of queries. In our specific example, the application modeler must consider three interesting input fields (client, schedule, requirements) which lead to seven queries that may be of interest — more possibilities for input greatly amplify the problem. These difficulties that we have encountered has given us reason enough to reconsider and enhance our approach in order to achieve more user- and engineering friendliness. In the following, we show how to tackle these two issues together.

4. Proactive inferencing agents

The application that we have just outlined provides additional assistance for daily business tasks. However, it is still very far from your favorite secretary some of whose capabilities we have listed in Section 1, and its modeling requirements weigh heavily on the application modeler who realizes the reactive task support. In this chapter, we want to push the edge a little further by adding a flexible, proactive strategy to our application. For this purpose, we first describe the inference engine that is employed in a little more detail, before we go on to extend its capabilities.

4.1. Inference engine

Reasoning in the inference engine involves the following stages (cf. Ref. [8] for an elaboration):

1. The high-level modeling of ontology, facts and queries, which is all done in F-Logic, is translated into a set of horn clauses.²

² F-Logic is a frame-logic representation language conceived by Kifer et al. [10]. In the implementation by Angele and Decker that we use, F-Logic is a proper subset of first-order predicate logic. Concepts and relations are reified and, hence, may be treated as first-order objects over which quantification is possible. For efficient processing, F-Logic is translated into a datalog-style representation [5,14].

2. The horn clauses are processed using a strategy called dynamic filtering [8]. The implemented strategy may very well be conceptualized by a dynamic programming approach with an agenda as its central data structure (cf. Ref. [4]). This means that inferencing sub-tasks are put onto the agenda, ordered according to a particular strategy, and then retrieved from the agenda for actual processing.³
3. The inference engine — in contrast to simple Prolog interpreters — allows to store the facts that have once been derived in the database. Hence, if the same or similar queries are posed repeatedly, the computational load may be much lower than if the derivation must be started from scratch.
4. The variable bindings that were open in the query are returned by the inference engine, thus, yielding a set of tuples corresponding to all the facts in the semantic model that match the query.

Thus, the F-Logic inference engine by Angele and Decker combines ordering-independent reasoning in a high-level logical language with a well-founded semantics and a very flexible mechanism for modifying reasoning strategies that we will exploit in the following.

4.2. Proactive inferencing

Let us now reconsider the example setting described in the reactive framework. From the inferencing point of view the key components were: (i) an application that provides facts concerning the content and the context of a query; (ii) precompiled queries, which will be executed if all the related facts are given; and (iii) a deductive database that retrieves and deduces answers to queries.

Now, in order to proceed from a reactive to a proactive approach we must cope with: (i) an application that defines the context only partially, and with; (ii) semi-defined queries, where not all facts related to a query have been given yet; and (iii) we must provide a deductive database that takes advantage of pre-computations in order to react quickly.

As we have found these are not requirements that contradict each other, but indeed with the proper inferencing

³ For instance, a *last-in-first-out* strategy leads to a depth-first, a *first-in-first-out* to a breadth-first processing strategy.

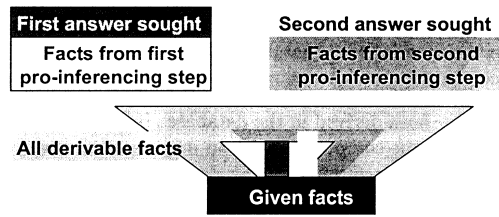


Fig. 4. Model precomputation through pre-inferencing.

strategy they complement each other quite nicely. The high level view onto the interaction of these needs is depicted in Fig. 4. At the beginning, we are given a set of facts and an ontology describing a set of concepts, relations, and rules. The facts that might be derived by applying all the rules in a forward-chaining manner would most often cause an overflow of data in the data storage — in fact the semantic model could even be an infinite one, as it may easily occur when functions are used in the representation language. On the other hand, we find that there is a small set of facts that is actually needed as background information for the knowledge worker. For instance, in one situation the knowledge sought from the database might be sketched in the form of a small rectangle (cf. “First answer sought”). Since neither the context nor the actual background knowledge is completely specified, the best one can hope from the inference engine is a set of facts that contains the knowledge required at this point of work, but that is not minimal with regard to the actual task the knowledge worker has to solve.

However, this need not even be a disadvantage. In fact, we have described above that formerly derived facts need not be recomputed by the inference engine, thus, the pre-computation of a reasonably small set of facts may even be useful when one moves on to another setting and another, similar, task. Just think of a task where you must contact several people that are related to a particular topic, e.g. knowledge management. Their actual relationship to knowledge management may be computed from facts like “person A has written a paper about knowledge management”, “person B has written a mail to me about knowledge management”, etc. In a particular context you might wish to contact person B, because you are co-authoring a paper — and of course, it is much more likely that you co-author a paper with a person with whom you have contact than with a person whom you might not know at all. The inference engine will probably fail to employ this rationale, since its contextual knowledge is very limited. But when you try to contact person A later on anyway (say in order to organize a workshop) then the inference engine may take full advantage of previous computations — the second answer you seek is already contained in the first set of facts that have been computed before (Fig. 4).

Now the question remains open as to how the strategy just explained is realized in the actual application. The realization must take into account that model pre-computation is expensive and that the user may pose an explicit query at

any point in time. Hence, one needs to avoid that model pre-computations overflow the data storage with useless facts and that the pre-computation actually prevents efficient query answering. To account for these stipulations we apply the following scheme:

1. The application queries the system when it determines that enough facts that belong to a query are known to the agent. In our setting this currently means that at least one open variable must be bound to a concrete fact, but in other settings more restrictive strategies might be necessary.
2. We distinguish between three different query priorities, viz. explicit queries by the user (priority 1), queries the variable of which are completely bound by the application (priority 2), and queries with unbound variables (termed “underspecified queries”; priority 3). Underspecified queries receive the lowest priority, they are preempted by the other two types of queries and they are terminated when their computation reaches a time limit. Explicit queries by the user receive the highest priority. They preempt priority 2 queries such that the computation of the latter may be continued after the completion of the explicit query.
3. To allow for preemption and (if necessary) continuation of a query, the agenda strategy is modified. Entries on the agenda that are of priority 3 are cleared from the queue when a query of priority 1 or 2 arrives or when the deadline for a query expires. Entries on the agenda that are of priority 1 are put at the beginning of the agenda such that they are handled first, but such that the processing of entries of priority 2 may be resumed after their completion. By this way, the inference engine is adapted to yield an anytime algorithm [4].
4. Queries with several unbound variables easily lead to an abundance of possible results, thus, deteriorating the specificity that one might gain from our approach compared to information retrieval-based schemes. Hence, it is important to restrict actual output to results that can be easily grasped by the user. This means we only provide hints proactively, if the result is definite, i.e. if in spite of the underdefined context and background knowledge only one possible answer may be found for a particular entry. Furthermore, we provide a selection of up to seven possible choices, if we expect the user to continue with just the information that we have computed in our approach.

All in all, this scheme allows the provision of the power that we have sketched above: it takes full advantage of pro-active inferencing through a multiple-priority approach. Pre-computation comes with an internal time limit, such that we gain a careful balance between the derivation of definite results from a weakly defined context, the pre-computation of facts queried later, and the computational loads for processing and storage required from the system.

4.3. Example for proactive processing

As mentioned in Section 3.4, it is nearly impossible to model both, enough queries such that you get the answer that you are looking for and not too many queries such that the user and the modeler can both deal with the number of queries that are required for a particular task.

In contrast, an intelligent assistant adapts to the particular facts it finds in its knowledge base. It employs specific queries when it is appropriate to find out about who of your colleagues fits best into your work plan and it employs general queries when a too specific one would lead you into a dead end road — our proactive scheme is appropriate to realize just that. For instance, resuming our running example, a project manager may want to include someone with XML expertise. Assuming content in the knowledge base like that given in Table 3, the necessity to include someone with XML expertise requires that Joe Doe is contacted for a fruitful project. The problems in the reactive framework come from the fact that a specification like ‘an expert in “XML” who is available from May–June 2010’ is so specific that in many, many cases an empty set of corresponding facts will be derived.

In the proactive framework the situation is somewhat different. Let us assume that the project manager wants to contact two persons. One who knows about XML and one who is an expert with ERP Tool A. Furthermore, let us assume that the project manager deals with a template that queries for people depending on their expertise and availability during a scheduled time frame. When the manager first plans for the position requiring XML knowledge, she mentions this in her planning template. In the reactive framework she would have to enter the time schedule, too — and find out that no such person exists or she would have to choose from a possibly long menu of queries. In the proactive framework, the inference engine starts with a low priority query as soon as the stipulation for XML expertise has been entered to the system. Since only one expert for XML may be retrieved at all, the corresponding instance — possibly with some explanation — may be returned even before the project manager may start to plan more details. Thus, the agent uses possible idle times of the user and the user gets an early feedback on her plans that is collected from the organizational memory, the axioms in the ontology and the proactive inferencing scheme.

When the project manager plans for someone with expertise in “ERP Tool A”, there are two possibilities. If the number of experts is low, the agent returns the selection of experts in the field. Otherwise it does not react until the time schedule is specified such carefully that the choice is narrow enough to present it to the user. Hence, the careful balancing between overly specific and overly general questions is delegated to the agent and need not be handcrafted. In the reactive framework, an overly specific question might result in no information at all — yielding no information about Joe Doe, while an overly general question might

easily plague the user with an abundance of query lists that she does not want to cope with when she has only described a few specifications yet.

5. Related work

Our work is an organizational memory-based approach for proactive support of knowledge workers through an intelligent agent. In the research field of information agents, numerous systems for very specific domains and tasks exist and also several publications classify and describe those systems [11,21]. For example, Liebermann [13] gives an overview of approaches to design intelligent information agents (Letizia, Remembrance Agent, LetsBrowse, Firefly, Butterfly, ExpertFinder, Tet-a-Tete, Footprints System) that provide active assistance in the process of finding and organizing information. These agents learn from interaction with the user and anticipate the users needs. This happens through the analyses of statistical information of web pages and user profiles. In contrast, we embed our agent support in specific tasks of a workflow, defined by document structures that are related to ontological knowledge. So, we mainly differ from those information agents in that we use semantic, and not only statistical information.

In the knowledge management area related to our approach, the distinction is not so clear cut and the surveys are not so numerous. Hence, we here give a more detailed roadmap of these: Our starting point has been a common intranet environment, in which all documents are put such that they are widely available for reuse and general information. The next step has been an integration of distributed factual knowledge with an ontology as its conceptual backbone. Thereby, we relied on the system Ontobroker. Indeed, Benjamins et al. [3] already outlined how Ontobroker could be used for knowledge management. However, in their approach the user had to bear all the burden of doing the right things at the right time, while our approach goes in the direction of telling the user what might be useful for him in his very next task — still leaving the option for user-initiated queries. A central point in our approach is reasoning about document structure and contents. Here we take over Weitz’s view of SGML documents in a workflow process [22], but extend his approach to include the knowledge management side and the weakly structured parts of processes.

Nearest to our integration of workflow and knowledge management aspects comes research presented in Refs. [2,9,12,15,16]. Huber [9] builds on a Lotus Notes intranet environment that lets the user define a simple ontology and small workflows. However, his approach is less principled and does not lend itself easily for modeling and process planing goals. In particular, he cannot query facts, not to speak of implicit knowledge, but only documents.

Reimer et al. [16] support the user with particular tasks. For this purpose, they use rather rigid process structures that

are build from declarative business rules. We, in contrast, prepare appropriate solutions, but ultimately leave all the decisions as well as the ordering of business tasks to the user. We just try to advise him with information that may facilitate his problem solving.

Ackerman and Mandel [2] describe an approach that hierarchically structures tasks and abstracts from different types of data collections in order to support the users in their purpose of analyzing astronomical data. Thus, they pursue a goal that is comparable to ours. However, their application is much more dedicated to their particular goal. With our approach we intend to reach a higher degree of flexibility as far as the task goals are concerned and we provide active support for the business tasks.

Mahling and King [15] describe an intelligent planning system that supports goal-based workflow similar as in our approach. For this purpose, they also devised an elaborate agent architecture such that electronic or human workflow participants may easily cooperate. Their approach, however, lacks an adequate level of description for the knowledge in the documents. Hence, the knowledge base of their system does not grow with its use, such as we require for the typical knowledge worker.

We share many of the convictions we build on with Leake et al. [12]. In their approach, they also aim at seamless interaction in task-based knowledge management. They provide an integration of various knowledge sources and some “proactive” support. We put *proactive* between quotation marks, since they use an approach that is more like our *Reactive Agent Support* where *queries*, rather than appropriate *answers*, are compiled proactively. Also in contrast to our approach, they use information retrieval and case-based reasoning techniques. This may appear advantageous, because their system may perhaps degrade somewhat more gracefully when its is given over- or underspecific queries when compared to our reactive agent support. However, the drawback they incur is that their scheme is not semantically based and, hence, may not provide similar semantic rigorosity, semantic derivations, and semantic-based, proactive compilation of appropriate answers.

Our approach builds heavily on considerations by Abecker et al. [1] who establish a common ground for documents, organization and knowledge (reflected by their information, enterprise and domain ontologies, respectively). However, they do not exploit the full potential of ontologies, as they completely neglect inferencing issues that are a major topic in this paper.

We differ from common workflow management and office information support systems by considering the document semantics in detail. In particular, we model document structures in order to provide better process support through inferencing on document knowledge. This support is not restricted to rigidly structured processes, but it may easily be exploited for weakly structured processes, too, where only parts of the overall order of task decomposition are known.

6. Conclusion

In this paper we have presented an approach for intelligent, proactive inferencing agents that subsumes our earlier work on desk support that combined an organizational memory with business process modeling [19]. The reasons for the extension of the earlier approach stem from a comparison of the agent with a (very roughly) corresponding human assistant. Our first approach exhibited a lack of flexibility and proactivity in adding help to the project management setting or comparable tasks, while in the improved scheme presented here, the modeling of interesting queries is facilitated. This is because careful balancing between overly specific and overly general questions may be delegated to the agent and need not be hand crafted as before.

In this paper, we have again focused on the problem of project management. However, this does not mean that we are restricted to this scenario. As one may easily see, the — possibly digital — travel agent that supports you in your booking of your next vacation trip has to deal with the very same problems. The task requires a comprehensive memory with an ontological structure for all different types of housing and transport. An intelligent, supportive agent would proactively think about what you are doing. If you try to get a flight during Christmas season it might proactively determine that no matter what airline or exact date you choose, there are just no flights available for the cheapest airfare anymore. Current systems badly lack this type of proactive reasoning, as one of the authors had to experience recently. The importance of such an approach may then be derived by the time a particular project manager saves in executing such a task, the more impressive and important consequences, however, will be derived from customer satisfaction as one of the key factors in enterprise services.

Acknowledgements

We thank our colleagues in the knowledge management research group for many fruitful discussions and Jürgen Angele and Stefan Decker from Ontoprise GmbH for their support in using their inference engine. This work was partially funded by BMBF under grant “01IN802” (project “GETESS”) and by the European Union for the research project “On-To-Knowledge”.

References

- [1] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, M. Sintek, Toward a technology for organizational memories, *IEEE Intelligent Systems* 13 (3) (1998) 40–48.
- [2] M. Ackerman, E. Mandel, Memory in the small: combining collective memory and task support for a scientific community, *Journal of Organizational Computing and Electronic Commerce* 1999, 9 (2–3), pp. 105–127.
- [3] V.R. Benjamins, D. Fensel, A. Gómez Pérez, Knowledge management through ontologies, in Ref. [17], pp. 5.1–5.12.

- [4] M.S. Boddy, Anytime problem solving using dynamic programming, AAAI-91 — Proceedings of the National Conference on Artificial Intelligence 2 (1991) 738–743.
- [5] S. Decker, On domain-specific declarative knowledge representation and database languages, in: A. Borgida, V. Chaudri, M. Staudt (Eds.), KRDB-98 — Proceedings of the Fifth Workshop Knowledge Representation Meets DataBases, WA Seattle, 31 May 1998.
- [6] S. Decker, M. Erdmann, D. Fensel, R. Studer, Ontobroker: ontology based access to distributed and semi-structured information, Database Semantics, Semantic Issues in Multimedia Systems, Kluwer, Boston, MA, 1999, pp. 351–369.
- [7] M. Erdmann, R. Studer, How to structure and access XML documents with ontologies. Data and Knowledge Engineering (Special Issue on Intelligent Information Integration), (in press).
- [8] D. Fensel, J. Angele, R. Studer, The knowledge acquisition and representation language KARL, IEEE Transactions on Knowledge and Data Engineering 10 (4) (1998) 527–550.
- [9] H. Huber, Document research based on collaborative provided structure knowledge, in Ref. [17], pp. 11.1–11.9.
- [10] M. Kifer, G. Lausen, J. Wu, Logical foundations of object-oriented and frame-based languages, Journal of the ACM (1995) 42.
- [11] M. Klusch (Ed.), Intelligent Information Agents — Agent-Based Information Discovery and Management on the Internet Springer, Berlin, 1998.
- [12] D. Leake, L. Birnbaum, K. Hammond, C. Marlow, H. Yang. Task-based knowledge management, in: Exploring Synergies of Knowledge Management and Case-Based Reasoning, Proceedings of the AAAI-99 Workshop, AAAI 1999, pp. 35–39.
- [13] H. Liebermann. Personal assistants for the web: an MIT perspective, in: Ref. [11], pp. 279–292.
- [14] J.W. Lloyd, R.W. Topor, Making Prolog more expressive, Journal of Logic Programming 1 (3) (1984).
- [15] D.E. Mahling, R.C. King, A global-based workflow system for multiagent task coordination, Journal of Organizational Computing and Electronic Commerce 9 (1) (1999) 57–82.
- [16] U. Reimer, A. Margelisch, B. Novotny, T. Vetterli, Eule2: a knowledge-based system for supporting office work, CoopIS-99: Proceedings of the Fourth International Conference on Cooperative Information Systems, UK Edinburgh, September 1999.
- [17] U. Reimer (Ed.), Proceedings of the Second International Conference on Practical Aspects of Knowledge Management, Basel, Switzerland, 29–30 October 1998.
- [18] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Mädche, H.-P. Schnurr, R. Studer, Y. Sure, Semantic community web portals, Computer Networks (Special Issue: WWW9 — Proceedings of the Ninth International World Wide Web Conference, Amsterdam, The Netherlands, 15–19 May 2000), pp. 473–491.
- [19] S. Staab, H.-P. Schnurr. Knowledge and business process: approaching an integration, in: R. Dieng, S. Decker, N. Matta, U. Reimer (Eds.), Proceedings of the IJCAI-99 Workshop on Knowledge Management and Organizational Memory, Stockholm, July 31–August 6 1999.
- [20] W3C, Extensible markup language (xml), <http://www.w3.org/XML/>, 1997.
- [21] G. Weiss, Multiagent Systems — A Modern Approach to Distributed Artificial Intelligence, MIT Press, MA Cambridge, 1999.
- [22] W. Weitz, Combining structured documents with high-level petri-nets for workflow modeling in internet-based commerce, Journal of Cooperative Information Systems 7 (4) (1998) 275–296.

Lebenslauf Steffen Staab

Zentrale Daten

Name Steffen Rudolf Staab

Titel Dr. rer. nat., Master of Science in Engineering (University of Pennsylvania)

Geboren 3 Januar 1970 in Würzburg

Kinder Eine Tochter, geb. 2000

Ausbildung

1976-1980 Grundschule, Zellingen

1980-1989 Gymnasium, Karlstadt am Main; Abitur mit Note 1,1

1990-1993 Studium der Informatik an der Universität Erlangen-Nürnberg

1993-1994 Studium von Computer and Information Science an der University of Pennsylvania, Philadelphia, PA, USA

1994 Master of Science in Engineering (Computer and Information Science), Univ. of Pennsylvania, Philadelphia, USA; Thesis "GLR-Parsing of Word Lattices" (Betreuer: Prof. Dr. Mark Steedman, Philadelphia; Prof. Dr. Günter Görz, Erlangen)

1995-1998 Promotionsstudium im Graduiertenkolleg "Menschliche und Maschinelle Intelligenz" und in der Computerlinguistik der Universität Freiburg

1998 Dr. rer. nat. in Informatik, Universität Freiburg
Doktorarbeit "Grading Knowledge - Extracting Degree Information From Texts". Gutachter: Prof. Dr. Hahn, Prof. Dr. Ottmann; Dissertationskomitee: Prof. Dr. Nebel, Prof. Dr. Basin. Note: Sehr Gut

2001/2002 Antrag auf Habilitation an der Universität Karlsruhe (TH) eingereicht. Habilitationsverfahren ist eröffnet. Gutachter: Prof. Dr. Studer, Prof. Dr. Waldmann, Prof. Dr. Lockemann (alle: Univ. Karlsruhe), Prof. Dr. Jarke (RWTH Aachen). Vorläufiger Termin des Habilitationsvortrags: 29. Mai 2002

Stipendien / Auszeichnungen

1991-1995 Stipendium der Studienstiftung des Deutschen Volkes

1992 Auszeichnung für das beste Vordiplom der Erlanger Informatik des Jahrgangs 1990 (ca. 260 InformatikstudentInnen)

- 9/1993-5/1994 Fulbright-Stipendium für Studium an der University of Pennsylvania
- 1995-1998 Stipendium im Graduiertenkolleg "Menschliche und Maschinelle Intelligenz" der Universität Freiburg, Förderung durch die Deutsche Forschungsgemeinschaft (DFG)
- 1998 Best Paper Award der *European Conference on Artificial Intelligence, ECAI 1998*, für S. Staab. On Non-binary Temporal Relations
- 1999/2000 Short term research fellowship (6 Wochen) bei British Telecom Laboratories, Ipswich, UK
- 2001 Erster Preis der Integrata-Stiftung, Tübingen, (Dr. Wolfgang Heilmann - Preis für humane Nutzung der Informationstechnologie) für die Forschungsgruppe „Wissensmanagement“ (R. Studer, M. Erdmann, A. Mädche, H. Oppermann, H.-P. Schnurr, S. Staab, C. Tempich, Y. Sure)
- 2002 Best Paper, 2nd Prize, Award der 9th *International Conference for Information and Communication Technologies in Tourism, ENTER 2002* für A. Mädche & S. Staab. Applying Semantic Web Technologies for Tourism Information Systems

Berufserfahrung / sonstige Tätigkeiten

- 6/1989-8/1990 Grundwehrdienst
- 1991 4 Monate Werkstudent bei Software Design & Management (SD&M), München
- 6/1998-11/1998 Consultant am Fraunhofer Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart, im Bereich Softwaretechnik (Themenbereiche: Wissensmanagement, Usability Engineering)
- 11/1998-3/2000 Projektleiter (BAT Ib) am Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), Universität Karlsruhe (TH), in der Forschungsgruppe „Wissensmanagement“ von Prof. Dr. Studer
- Seit 6/1999 Mitgründer, Teilhaber und wissenschaftlicher Beirat der Firma Ontoprise GmbH, Karlsruhe, <http://www.ontoprise.de>
- Seit 4/2000 Wissenschaftlicher Assistent (C1) am Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), Universität Karlsruhe (TH), in der Forschungsgruppe „Wissensmanagement“ von Prof. Dr. Studer
- 8/2000 Eröffnung des Ontoprise Büros in der Technologiefabrik Karlsruhe
- 6/2001 Finanzierung der Ontoprise GmbH durch Venture Capital

Lehrerfahrung

- Lehrauftrag Dozent im Master-Studiengang "Wissensmanagement" der TU Chemnitz (Koordination: Prof. Dr. Pawlowsky), WS 2002/2003
- Vorlesungen "Intelligente Systeme auf dem Web" (SS 2002)
"Knowledge Discovery" (WS 2001/2002)
"Intelligente Systeme auf dem Web" (SS 2001)
"Knowledge Discovery" (WS 2000/2001)
"Intelligente Systeme auf dem Web" (SS 2000)
"Knowledge Discovery" (WS 1999/2000)
- Seminare Seminar "Wissensbasierte elektronische Märkte" (SS 2002)
Seminar "E-Business & Intelligent Web" (WS 2001/2002)
Seminar "Semantic Web" (SS 2001)
Projektseminar "Semantic Web" (WS 2000/2001)
Projektseminar "Knowledge Discovery" (SS 2000)
Projektseminar "Text Mining" (WS 1999/2000)
Projektseminar "Wissensmanagement" (SS 1999)
Seminar "Programmieren in Lisp" (SS 1996)

Eingeladene Vorträge

Eingeladener Vortrag "Multiple Use of Ontologies in Functional Genomics" (vorläufiger Titel), Workshop "Ontology for Biology", European Media Lab, Heidelberg, November 7-8, 2002 (Part of the programme "Integrated Approaches to Functional Genomics")

Eingeladener Vortrag "Semantic Community Web Portals" – Gaining Insight from Research Information – *6th International Conference on Current Research Information Systems*. Kassel, Germany, August 29 - 31, 2002

Keynote, "Ontology Engineering", *Fifth International Baltic Conference on DB and IS*, Tallinn, Estonia, June 3-6, 2002

Eingeladener Vortrag „Machine Learning and the Semantic Web“. Benelearn 2001, the 11th Dutch-Belgian conference on Machine Learning, Antwerp, Belgium, December 21, 2001

Eingeladener Vortrag "Metadata and Semantic Web". DELOS Workshop on Interoperability in Digital Libraries. Workshop at ECDL-2001. GMD-IPSI, Darmstadt, Germany, September 8-9, 2001

Eingeladener Vortrag "Knowledge Portals – Taking advantage of Human Language Technology" 2001 EACL/ACL Workshop on Human Language Technology and Knowledge Management (HLT & KM), Toulouse, July 6-7 at ACL '01

Eingeladener Vortrag "Knowledge Portals". Auf dem zweiten HERMES (Human network of research and development in the field of advanced learning environments over the WWW) Workshop über "Knowledge-based systems for the delivery and adaptation of educational courseware", Patras, Greece, 17 Februar 2001

Vorträge und Tutorien

Tutorial "Ontologies: Representation, Engineering, Learning and Applications" mit A. Mädche at the European Conference on Artificial Intelligence – ECAI-2002, Lyon, France, July, 21, 2002

Tutorial "Ontologies: Representation, Engineering, Learning and Applications" mit A. Mädche at the 1st International Semantic Web Conference – ISWC 2002, Chia, Sardinia, Italy, June 10, 2002

Vortrag "Handling Metadata" mit S. Handschuh, W. Nejdl. Developers Day at WWW2002, Honolulu, Hawaii, May 11, 2002

Gastvortrag in der Vorlesung „Künstliche Intelligenz II“ von Prof. Nejdl, Hannover (per Videokonferenz aus Karlsruhe), 30 April 2002

Vortrag "Technologische Grundlagen des Semantic Web", 9. AIK-Symposium, Thema „Semantic Web“ Karlsruhe, 19 April, 2002 (Verein Angewandte Informatik Karlsruhe e.V.)

Vortrag "Semantische Technologien und Peer-to-Peer für das Wissensmanagement" mit J. Angele am 9. CNEC- Symposium "Geteiltes Wissen ist doppeltes Wissen – Möglichkeiten und Grenzen von Peer-to-Peer- (P2P-) Lösungen" am 27 Februar 2002 (Competence Network Electronic Commerce)

Vortrag "SEAL – Tying Up Information Integration and Web Site Management by Ontologies", IRST Trento, 27 März 2002

Vortrag "Semantic Patterns", Dagstuhl Seminar zu „Rule Markup Techniques“, Dagstuhl, 6 Februar 2002

Vortrag "Ontology Learning", Univ. Zürich, 10 Januar, 2002

Vortrag "Ontology Acquisition", Univ. Stuttgart, Institut für maschinelle Sprachverarbeitung, 23 November 2001

IJCAI-2001 Tutorium "AI Techniques for Knowledge Management" mit S. Decker, Seattle, WA, USA, 5 August 2001

Vortrag "Ontology Learning", Univ. Heidelberg, Lehrstuhl für Computerlinguistik, 11 Juli 2001

Tutorium "Wissensdatenbanken". Im Rahmen des Seminars "Certified Information Manager" der Management Circle GmbH, Frankfurt, 21 Juni 2001

Vortrag "Ontology Learning". An der Ecole Nationale Supérieure des Télécommunications, Brest, Bretagne, France, 14 Juni 2001

Tutorium "Wissensdatenbanken für den Help Desk" auf dem Workshop "Wissensmanagement in der IT" veranstaltet von Management Circle GmbH, Frankfurt/Oberursel, 13 Dezember 2000

Tutorium "Wissensdatenbanken für den Help Desk" auf dem Workshop "Wissensmanagement in der IT" veranstaltet von Management Circle GmbH, München, 23 Januar 2001

ECAI-2000 Tutorium "AI Techniques for Knowledge Management" mit S. Decker, Berlin, 21 August 2000

Dagstuhl Vortrag "Ontology-based Knowledge Management", Dagstuhl Seminar on Knowledge Management, Dagstuhl, 11 Juli 2000

Vortrag "Intelligente Techniken für das Wissensmanagement" auf dem 5. AIK Symposium "Wissensmanagement", Karlsruhe, 5 Mai 2000

Vortrag "Ontologies for Knowledge Management". Seminar der Knowledge Management Research Group, British Telecom, Ipswich, UK, 20 Januar 2000

Weitere Vorträge

20 Vorträge über Konferenz- und Workshopbeiträge auf internationalen und nationalen Workshops und Tagungen (vgl. Publikationsliste)

Projektakquisition

Hauptautor und Koordination SWAP – Semantic Web and Peer-to-Peer. EU IST Key Action III, Action Line on Semantic Web. Gesamtumfang ca. 436 Personenmonate, davon 106 für Koordinator. Start 1.4.2002. Laufzeit 30 Monate.

Hauptautor und Koordination (Einreichung) LEON – Learning Ontologies: Automatic Construction, Coordination and Evolution of Multiple Ontologies. Hauptautorschaft und Koordination durch S. Staab. Positiv begutachteter Kurzantrag. Vollantrag eingereicht am 18. Februar 2002

Hauptautor OntoWise (DFG Projekt); 4/2000 – 3/2002

Koautor Personalized Access to Distributed Learning Repositories – PADLR (BMBF Projekt); 9/2001 – 12/2003

WonderWeb (EU IST Projekt); ca. 10/2001 – 3/2004

Skill Management (Kooperation mit DaimlerChrysler, Würth); Seit 11/2001 am FZI Karlsruhe

OntoLogging (EU IST Projekt); 1/2001 – 12/2003

Candle – Collaborative And Network Distributed Learning Environment; Subcontracting des AIFB 1/2001 - 4/2001

GETESS Phase 2 (BMBF); 4/1999 – 6/2001

OntoAgents (DAML, DARPA Projekt); Start 9/2000

OnToKnowledge (EU IST); 2/2000 – 7/2002

Invite (BMBF Leitprojekt am Fraunhofer IAO); 2000-2002

Projekterfahrung

10/1992-12/1992 und 7/1994-12/1994 Studienarbeiter und Hilfswissenschaftler in einem Teilprojekt des BMBF-Projektes „Verbomobil“ in Erlangen, Konzeption und Programmierung von Parsingalgorithmen

6/1995-5/1998 Mitarbeit und Konzeption am Textverstehenssystem „Syndikate“ der Arbeitsgruppe Computerlinguistik der Universität Freiburg

7/1998-11/1998 Softwareprojekt für das Call-Center eines großen deutschen Versicherers, Zuständigkeit für Usability

11/1998-6/2001 BMBF-Projekt „GETESS – German Text Exploitation and Search System“ – Leitung des Karlsruher Teilprojekts zur Entwicklung von Ontologien und Ontologiewerkzeugen

Seit 6/1999 Mitarbeit und Projektleitung für die Entwicklung von ontologiebasierter Software und Semantic Web Anwendungen für Ontoprise GmbH, Karlsruhe

8/1999-1/2000 Beratung bei Entwicklung und Konstruktion einer ontologiebasierten Intranetlösung (Industrieprototyp) für Multiprojektmanagement bei British Telecom, Ipswich, UK, im Rahmen einer Short Term Research Fellowship

Seit 1/2002 EU IST FET-O Projekt „WonderWeb“, Projektleitung des Karlsruher Teilprojektes

Seit 4/2002 EU IST Projekt „Bizon“, Projektleitung des Karlsruher Teilprojektes bei Ontoprise

Weitere Mitarbeit in den Projekten

- EU-Projekt „OnToKnowledge“ – Ontology-based Knowledge Management, seit 1/2000
- DARPA DAML-Projekt „OntoAgents“ – Infrastructure for the Semantic Web, seit 8/2000
- BMBF „Personalized Access to Distributed Learning Repositories“, Learning Lab Lower Saxony, Hannover, seit 9/2001

Organisatorische Tätigkeiten

- Editor Department Editor „Trends & Controversies“, IEEE Intelligent Systems, seit 11/2001
- Co-chair Workshopcochair “WM2003 – Professionelles Wissensmanagement – Erfahrungen und Visionen“, mit G. Stumme, Luzern, Schweiz, 2-4 April 2003
- “Knowledge Markup and Semantic Annotation“ mit S. Handschuh, R. Dieng, N. Collier. Workshop at ECAI 2002, Lyon, France, August 2002
- SemanticWeb-2002 – Workshop on the Semantic Web (auf der WWW-2002 Konferenz), mit N. Fridman Noy und M. Frank, Hawaii, USA, 7 Mai 2002
- Fachgruppentreffen der FG Wissensmanagement, “German Workshop on Experience Management“, mit M. Minor, Berlin, 7-8 März 2002
- “Knowledge Markup and Semantic Annotation“, mit S. Handschuh, R. Dieng. Workshop at K-Cap 2001, Vancouver, USA, October 2001
- Workshop on "Ontologies" (auf der KI-2001) mit A. Mädche, G. Stumme, Vienna, Austria, September 19-21, 2001
- OL-2001 – Second Workshop on Ontology Learning (auf der IJCAI-2001), mit A. Mädche, Seattle, WA, USA, 4 August 2001
- SemanticWeb-2001 – Workshop on the Semantic Web (auf der WWW-10 Konferenz), mit S. Decker, D. Fensel, A. Sheth, Hongkong, China, 1 Mai 2001
- WM2001 – Professionelles Wissensmanagement – Erfahrungen und Visionen. Mit R. Studer, Baden-Baden, 14-16 März 2001
- OL-2000 – ECAI-2000 Workshop on Ontology Learning, mit A. Mädche, Berlin, 22 August 2000
- AAAI Spring Symposium 2000 “Bringing Knowledge to Business Processes“ mit D. O’Leary, Stanford, CA, USA, 20-22 März 2000
- Metadata Chair 1st International Semantic Web Conference (ISWC-2002), Sardinia, Italy, 10-12 June, 2002
- Organisations-“Learning and Text Analysis for Ontology Engineering“,
kommitteemitglied Workshop at ECAI 2002, Lyon, France, August 2002
- OM-2000 – ECAI-2000 Workshop on “Knowledge Management and Organizational Memories“, Berlin, 21 August 2001

Reviewing

- Programm- 4th International Conference Practical Aspects of Knowledge
kommitteemitglied Management. Vienna, Austria, 2-3 December 2002
- Colloque International sur la Fouille de Texte – CIFT-2002.
Hammamet, Tunisie, October 20-23, 2002
- Workshop on Knowledge Management through Corporate
Semantic Webs at EKAW'2002, Sigüenza, Spain, October 2002
- Workshop on Web Semantic (WEBS) at DEXA-2002. Aix-En-
Provence, France, 2-6 September, 2002
- International Workshop on Natural Language and Information
Systems (NLIS 2002), Aix-en-Provence (France), September 2-
6, 2002
- World Computer Congress - WCC 2002 Stream on "Intelligent
Information Processing" (IIP 2002), Montreal, Canada, August
25-30, 2002
- STAIRS – European Starting AI Researcher Symposium (at
ECAI2002), Lyon, France, July 21-26, 2002
- Workshop on Knowledge Management and Organizational
Memories (at ECAI2002), Lyon, France, July 21-26, 2002
- Workshop on Ontologies (at ECAI2002), Lyon, France, July 21-
26, 2002
- GI Workshop "XML Technologien für das Semantic Web". Berlin,
24-25 Juni 2002
- 1st International Semantic Web Conference (ISWC-2002),
Sardinia, Italy, 10-12 June, 2002
- FLAIRS 2002 (Special Track on Semantic Web), Pensacola,
Florida, May 16-18, 2002
- WWW11 – 11th International Conference on the World Wide
Web, Hawaii, May 6-11, 2002
- International Conference on Intelligent User Interfaces -
IUI 2002. San Francisco, CA, USA, January 27-30, 2002
- Workshop on Integrating Data Mining and Knowledge
Management. November 29, 2001 at ICDM'01: The 2001 IEEE
International Conference on Data Mining, San Jose, California,
USA, November 29 - December 2, 2001
- WI-IF 2001 - 5. Internationale Tagung Wirtschaftsinformatik
und 3. Tagung Informationssysteme in der Finanzwirtschaft. Im
Track on the Semantic Web and E-Business, Augsburg, 19-21
September 2001
- Workshop on Theory and Applications of Knowledge
Management - TAKMA 2001. Workshop auf der DEXA 2001,
München, 3-7 September 2001

International Semantic Web Working Symposium (SWWS)
"Infrastructure and Applications for the Semantic Web",
Stanford, CA, USA, July 30-August 1, 2001

Context 2001 – Third International and Interdisciplinary
Conference on Modeling and Using Context, Dundee, Scotland,
July 27-30, 2001

IJCAI-2001 Workshop on "Ontologies and Information Sharing",
Seattle, WA, USA, 4-5 August 2001

"Individual and Organizational Learning for Software
Improvement" (INOR '01) auf der 13th Int. Conf on Software
Engineering and Knowledge Engineering (SEKE '01), Buenos
Aires, Argentina, 12 Juni 2001

Geschäftsprozessorientiertes Wissensmanagement – Von der
Strategie zum Content. Workshop auf der deutschen Konferenz
"WM2001", Baden-Baden, 14-16 März 2001

Knowledge Management by Case-Based Reasoning: Experience
Management as Reuse of Knowledge (GWCBR 2001). Workshop
auf der deutschen Konferenz "WM2001", Baden-Baden, 14-16
März 2001

PAKM 2000 - Third International Conference on Practical
Aspects of Management. Basel, CH, 30-31 Oktober

ECAI-2000 Workshop on "Ontologies and Problem-Solving
Methods", Berlin, 21 August 2000

First International Workshop on Theoretical and Practical
Aspects of Knowledge Management – TAPAKM 2000 (auf der
DEXA-2000), London, UK, 6-8 September 2000

COLING-2000 Workshop on "Semantic Annotation and
Intelligent Content", Centre Universitaire, Luxembourg, 5/6
August 2000

Sonstiges Reviewing:

- Zeitschriften
- IEEE Transactions on Circuits and Systems for Video Technology – Special Issue on Conceptual and Dynamical Aspects of Multimedia Content Description (5/2002).
 - IEEE Computer (3/2002)
 - International Journal of Human Computer Studies – IJHCS (3/2002)
 - Decision Support Systems (12/2001)
 - European Journal of Information Systems (11/2001)
 - Distributed and Parallel Databases (11/2001)
 - Autonomous Agents and Multi-Agent Systems (9/2001)
 - IEEE Intelligent Systems (12/2000)
 - KAIS: Knowledge and Information Systems (12/2000)
 - Transactions on Knowledge and Data Engineering (6/2000)
 - Data & Knowledge Engineering (12/99)
 - International Journal of Human-Computer Studies - IJHCS (12/98)
 - Constraints (8/98)
 - Journal of AI Research (8/98)
- Konferenzen
- EKAU 2002
 - KR 2002
 - ECAI 2002
 - K-Cap 2001
 - SEKE 2001
 - CAiSE 2001
 - IFIP Technical Committee TC12's Int. Conf. On Intelligent Information Processing (IIP-2000)
 - HICSS 2000
 - EKAU '99
 - CAiSE '99
 - Automated Software Engineering '98
- Workshops
- Modellierung 2000
 - Modellierung 1999
 - Intelligent Information Integration (at IJCAI-99)
 - Learning Software Organizations (at SEKE '99)

Schriftenverzeichnis von Steffen Staab

Buch

1999

- [1] ~. *Grading Knowledge – Extracting Degree Information from Texts*. LNCS/LNAI 1744. Heidelberg, Berlin: Springer Verlag, December 1999.

Zeitschriftenbeiträge

Angenommen zur Veröffentlichung

- [2] ~. Wissensmanagement mit Ontologien und Metadaten. *Informatik Spektrum*. Springer, 2002.

2002

- [3] A. Hotho, A. Mädche, ~. Text Clustering Based on Good Aggregations. *Künstliche Intelligenz*. Nr. 2, 2002.
- [4] A. Mädche, ~, R. Studer, Y. Sure, R. Volz. SEAL — Tying Up Information Integration and Web Site Management by Ontologies. *IEEE Data Engineering Bulletin*, March 2002: 10-17.

2001

- [5] ~. From Binary Temporal Relations to Non-Binary Ones and Back. *Artificial Intelligence*, 128(1-2): 1-29, Elsevier 2001.
- [6] ~, H.-P. Schnurr, R. Studer, Y. Sure. Knowledge Processes and Ontologies. *IEEE Intelligent Systems*, 16(1): 26-35, January/February 2001 (Special issue on knowledge management).
- [7] A. Mädche, ~. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2): 72-79, March/April 2001 (Special issue on the Semantic Web).
- [8] ~, A. Mädche. Knowledge Portals - Ontologies at Work. *AI Magazine*, 22(2), Summer 2001, pp. 63-75.
- [9] A. Hotho, A. Mädche, ~, R. Studer. SEAL-II — The soft spot between richly structured and unstructured knowledge. In *Journal of Universal Computer Science*, 7(7): 566-590. Springer, 2001.
- [10] M. Erdmann, A. Mädche, H.-P. Schnurr, ~. From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools. In *ETAI Journal - Section on Semantic Web (Linköping Electronic Articles in Computer and Information Science)*, 6(2), 2001.
- [11] ~, M. Erdmann, A. Mädche. Ontologies in RDF(S). In *ETAI Journal - Section on Semantic Web (Linköping Electronic Articles in Computer and Information Science)*, 6(9), 2001.
- [12] M. Klettke, M. Bietz, I. Bruder, A. Heuer, D. Priebe, G. Neumann, M. Becker, J. Bedersdorfer, H. Uszkoreit, A. Mädche, ~, R. Studer. GETESS –

Ontologien, objektrelationale Datenbanken und Textanalyse als Bausteine einer semantischen Suchmaschine. *Datenbank-Spektrum*, 1(1): 14-24, 2001.

2000

- [13] ~, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Mädche, H.-P. Schnurr, R. Studer, Y. Sure. Semantic Community Web Portals. In: *WWW9 / Computer Networks (Special Issue: WWW9 - Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, May, 15-19, 2000)*, 33(1-6): 473-491. Elsevier, 2000.
- [14] ~, H.-P. Schnurr. Smart Task Support through Proactive Access to Organizational Memory. *Knowledge-based Systems*, 3(5): 251-260. Elsevier, 2000.
- [15] E. Tsui, B. Garner, ~. The Role of Artificial Intelligence in Knowledge Management. *Knowledge-based Systems*, 13(5): 235-239. Elsevier, 2000.

Konferenzbeiträge

2002

- [16] A. Mädche, ~. Applying Semantic Web Technologies for Tourism Information Systems. In: Karl Wöber, Andrew Frew, Martin Hitz (eds.), *Proceedings of the 9th International Conference for Information and Communication Technologies in Tourism, ENTER 2002*. Springer Verlag, Innsbruck, Austria, 23 - 25th January 2002. **2nd Best Paper**.
- [17] T. Sollazzo, S. Handschuh, ~, M. Frank, N. Stojanovic. Semantic Web Service Architecture — Evolving Web Service Standards toward the Semantic Web. In: *Proceedings of the 15th International FLAIRS Conference*. Pensacola, Florida, May 16-18, 2002. AAAI Press.
- [18] S. Handschuh, ~. Authoring and Annotation of Web Pages in CREAM. In: *Proceedings of the 11th International World Wide Web Conference, WWW 2002*, Honolulu, Hawaii, May 7-11, 2002. ACM Press.
- [19] Y. Sure, M. Erdmann, J. Angele, ~, R. Studer, D. Wenke. OntoEdit: Collaborative Ontology Development for the Semantic Web. In: *Proceedings of the 1st International Semantic Web Conference (ISWC2002)*, June 9-12th, 2002, Sardinia, Italia. LNCS, Springer, 2002.
- [20] V. Pekar, ~. Taxonomy Learning – Factoring the structure of a taxonomy into a semantic classification decision. In: *Proceedings of the 19th Conference on Computational Linguistics, COLING-2002, August 24 – September 1, 2002*, Taipei, Taiwan, 2002.
- [21] ~, R. Studer, Y. Sure, R. Volz. SEAL – a SEmantic portAL with content management functionality. In: *Gaining Insight from Research Information. CRIS 2002 – Proceedings of the 6th Conference on Current Research Information Systems*, August 29-31, 2002, Kassel, Germany.

2001

- [22] A. Mädche, ~, N. Stojanovic, R. Studer, Y. Sure. SEAL – A Framework for Developing SEmantic portALs. In: Brian J. Read (Ed.): *Advances in Databases, Proceedings of the 18th British National Conference on Databases, BNCOD 18*, Chilton, UK, July 9-11, 2001. LNCS 2097, Springer Verlag, 2001, pp. 1-22.

- [23] N. Stojanovic, A. Mädche, ~, R. Studer, Y. Sure. SEAL — A Framework for Developing SEmantic PortALs. In: *K-CAP 2001 – Proceedings of the First International ACM Conference on Knowledge Capture*. October 21-23, 2001, Victoria, B.C., Canada.
- [24] S. Handschuh, ~, A. Mädche. CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework. In: *K-CAP 2001 – Proceedings of the First International ACM Conference on Knowledge Capture*. October 21-23, 2001, Victoria, B.C., Canada.
- [25] L. Stojanovic, ~, R. Studer. eLearning based on the Semantic Web. *WebNet 2001 - World Conference on the WWW and the Internet*. Orlando, Florida, USA, Oct. 23-27, 2001.

2000

- [26] ~, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Mädche, H.-P. Schnurr, R. Studer, Y. Sure. AI for the Web - Ontology-based Community Web Portals. In: *AAAI 2000/IAAI 2000 - Proceedings of the 17th National Conference on Artificial Intelligence and 12th Innovative Applications of Artificial Intelligence Conference*, Austin/TX, USA, July 30-August 3, 2000, Menlo Park/CA, Cambridge/MA, AAAI Press/MIT Press, pp. 1034-1039.
- [27] A. Mädche, ~. Discovering Conceptual Relations from Text. In: W.Horn (ed.): *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, Berlin, August 21-25, 2000. IOS Press, Amsterdam, 2000, pp. 321-324.
- [28] A. Mädche, ~. Semi-automatic Engineering of Ontologies from Text. In: *Proceedings of the Twelfth International Conference on Software Engineering and Knowledge Engineering (SEKE'2000)*. Chicago, July 6-8, 2000.
- [29] A. Mädche, ~: Mining Ontologies from Text. In: *R.Dieng & O Corby. EKAW-2000 - 12th International Conference on Knowledge Engineering and Knowledge Management*. October 2-6, 2000, Juan-les-Pins, France. LNAI, Springer.
- [30] S. Decker, J. Jannink, P. Mitra, ~, R. Studer, G. Wiederhold. An Information Food Chain for Advanced Applications on the WWW. In: *ECDL 2000 - Proceedings of the Fourth European Conference on Research and Advanced Technology for Digital Libraries*. Lisbon, Portugal, September 18-20, 2000. LNCS, Springer, 2000, pp. 490-493.
- [31] Y. Sure, A. Mädche, ~. Leveraging Corporate Skill Knowledge - From ProPer to OntoProper. In: *D. Mahling & U. Reimer. Proceedings of the Third International Conference on Practical Aspects of Knowledge Management*. Basel, Switzerland, October 30-31, 2000, pp. 22.1-22.9.
- [32] J. Angele, H.-P. Schnurr, ~, R. Studer. The Times They Are A-Changin' - The Corporate History Analyzer. In: *D. Mahling & U. Reimer. Proceedings of the Third International Conference on Practical Aspects of Knowledge Management*. Basel, Switzerland, October 30-31, 2000, pp. 1.1-1.11.

1999

- [33] ~, U. Hahn. Scalable Temporal Reasoning. In: *IJCAI '99 - Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Stockholm, Schweden, August, 1-6, 1999, pp. 1247-1252, San Francisco/CA: Morgan Kaufmann.

- [34] T. Barnekow, ~, J. Ziegler, R. Studer. An Architecture for Recovering Business Events Bottom-Up. In: *HCI '99 - Proceedings of the 8th International Conference on Human-Computer Interaction*, Munich, Germany, August, 22-27, 1999, pp. 614-618, London: Lawrence Erlbaum.
- [35] ~, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, B. Wrenger. A System for Facilitating and Enhancing Web Search. In: *Engineering Applications of Bio-Inspired Artificial Neural Networks. IWANN '99 - Proceedings of International Working Conference on Artificial and Natural Neural Networks*. Volume 2. Alicante, ES, June 2-4, 1999, pp. 706-714, LNCS 1607, Berlin, Heidelberg: Springer.
- [36] D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, ~, R. Studer, A. Witt: On2broker: Semantic-Based Access to Information Sources at the WWW. In: *Proceedings of the World Conference on the WWW and Internet (WebNet 99)*, Honolulu, Hawaii, USA, October 25-30, 1999.

1998

- [37] ~. On Non-Binary Temporal Relations. In: *ECAI '98 - Proceedings of the 13th European Conference on Artificial Intelligence*, Brighton, UK, August, 24-29, 1998, pp. 567-571, Chichester, UK: Wiley. **Best Paper**.
- [38] ~, U. Hahn. Grading on the Fly. In: *Cognitive Science '98 - Proc. of the 20th Annual Meeting of the Cognitive Science Society*, Madison, Wisconsin, August 1-4, 1998.
- [39] ~, U. Hahn. Distance Constraint Arrays: A Model for Reasoning on Intervals with Qualitative and Quantitative Distance. In: *AI'98 - Proceedings of the 12th Canadian Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada, June 18-20, 1998, LNAI 1418, pp. 334-348, Berlin: Springer.

1997

- [40] ~, U. Hahn. "Tall", "Good", "High" - Compared to What? In: *IJCAI '97 - Proc. of the 15th International Conference on Artificial Intelligence*, Nagoya/Japan, August 23-29, 1997, pp. 996-1001, San Francisco/CA: Morgan Kaufmann.
- [41] ~, U. Hahn. Comparatives in Context. In: *AAAI '97/IAAI '97 - Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference*, Providence/RI, USA, July 27-31, 1997, pp. 616-621, Menlo Park/CA, Cambridge/MA, AAAI Press/MIT Press.
- [42] ~, U. Hahn. Conceptualizing Adjectives In: G. Brewka, C. Habel & B. Nebel (Eds.), *KI '97 - Advances in Artificial Intelligence. Proc. of the 21st Annual German Conference on AI*, Freiburg, September 9-12, 1997, LNAI 1303, pp. 267-278. Berlin, Heidelberg: Springer.

1995

- [43] ~. GLR-Parsing of Word Lattices Using a Beam-Search Method. In: *Eurospeech-95: Proceedings of the European Conference on Speech Communication and Technology*. Madrid, September 1995.

Buchbeiträge

Angenommen zur Veröffentlichung

- [44] S. Handschuh, ~, A. Mädche. CREAting relational Metadata (CREAM) – A framework for semantic annotation. In: I. Cruz, S. Decker, J. Euzenat, & D. McGuinness (eds.) *The Emerging Semantic Web*. Selected papers from the Semantic web working symposium., IOS Press.
- [45] A. Mädche, ~, N. Stojanovic, R. Studer, Y. Sure. SEmantic portAL - The SEAL approach. In *Creating the Semantic Web*. D. Fensel, J. Hendler, H. Lieberman, W. Wahlster (eds.) MIT Press, Cambridge, MA.
- [46] A. Mädche, G. Neumann, ~. Bootstrapping an Ontology-based Information Extraction System for the Web. In: P.S. Szczepaniak, J. Segovia, J. Kacprzyk, L.A. Zadeh (eds.). *Intelligent Exploration of the Web*. Series "Studies in Fuzziness and Soft Computing". Springer/Physica-Verlag, Heidelberg.
- [47] ~, R. Studer, Y. Sure. Knowledge Processes and Meta Processes in Ontology-based Knowledge Management. C. Holsapple (ed.) *Handbook on Knowledge Management*. International Handbooks on Information Systems, Springer Verlag, 2002.
- [48] D. Fensel, ~, R. Studer, F. van Harmelen. Peer-2-Peer enabled Semantic Web for modern Knowledge Management. Davies, N.J., Fensel, D. & van Harmelen, F. (eds), *Ontology-based Knowledge Management: Exploiting the Semantic Web*, Wiley, London, UK, to appear Autumn 2002.
- [49] ~, H.-P. Schnurr. Knowledge and Business Processes: Approaching an Integration. In: R. Dieng & N. Matta. *Selected Papers from OM '99 to OM 2001*. Kluwer.

2001

- [50] ~, R. Studer. Intelligente (symbolische) Methoden für das Wissensmanagement. In: Radermacher et al. *Management von nicht-explizitem Wissen: Noch mehr von der Natur lernen*. Abschlussbericht - Teil 3 - Die Sicht verschiedener akademischer Fächer zum Thema des nicht-expliziten Wissens. FAW Ulm, pp 165-180.
- [51] ~, M. Erdmann, A. Mädche, S. Decker. An extensible approach for Modeling Ontologies in RDF(S). In R. Grütter (Ed.), *Knowledge Media in Healthcare: Opportunities and Challenges*. Hershey, USA: Idea Group Publishing, 2001, pp. 231-253.

2000

- [52] R. Studer, S. Decker, D. Fensel, ~. Situation and Perspective of Knowledge Engineering. In: J. Cuenca, Y. Demazeau, A. Garcia, J. Treur (eds.). *Knowledge Engineering and Agent Technology*. IOS Series on Frontiers in Artificial Intelligence and Applications, Volume 52. IOS Press, 2000.

Workshopbeiträge

2002

- [53] W. Nejdl, B. Wolf, ~, J. Tane. EDUTELLA: Searching and Annotating Resources within an RDF-based P2P Network. In: *Semantic Web Workshop at WWW-2002*, Honolulu, USA, May 7, 2002.
- [54] A. Hotho, A. Maedche, ~, V. Zacharias. On Knowledgeable Unsupervised Text Mining. In: *Proceedings of the DaimlerChrysler Workshop on Text Mining*. Ulm, April 26th-27th, 2002, to appear as Springer book.
- [55] S. Handschuh, ~, F. Ciravegna. S-CREAM – Semi-automatic CREAtion of Metadata. In: *SAAKM 2002 – Semantic Authoring, Annotation & Knowledge Markup. ECAI 2002 Workshop*, July 22-26, 2002, Lyon, France.

2001

- [56] S. Handschuh, ~, A. Mädche. CREAM: Creating relational metadata with a component-based, ontology-driven annotation framework. In *SWWS-1 – Proc. of the first Semantic Web working symposium*, Stanford, CA, US, July 29th-August 1st, 2001.
- [57] ~, A. Mädche, S. Handschuh. An Annotation Framework for the Semantic Web. In: S. Ishizaki (ed.), *Proc. of The First International Workshop on Multimedia Annotation*, January 30 - 31, 2001. Tokyo, Japan.
- [58] A. Mädche, ~. Learning Ontologies for the Semantic Web. In: *Proc. of Semantic Web 2001 – Workshop at WWW-10*, pp. 51-60, May 1, Hongkong, China.
- [59] ~, M. Erdmann, A. Mädche. Engineering ontologies using semantic patterns. In A. Preece & D. O’Leary (eds.), *Proceedings of the IJCAI-01 Workshop on E-Business & the Intelligent Web*. Seattle, WA, USA, August 5, 2001.
- [60] A. Hotho, A. Mädche, ~. Ontology-based text clustering. In *Proceedings of the IJCAI-01 Workshop on "Text Learning: Beyond Supervision"*. Seattle, WA, USA, August 6, 2001.

2000

- [61] ~, M. Erdmann, A. Mädche, S. Decker. An extensible approach for Modeling Ontologies in RDF(S). In: *Proceedings of the ECDL-2000 Workshop "Semantic Web: Models, Architectures and Management"*, Lisbon, September 21, 2000, pp. 43-52.
- [62] ~, A. Mädche: Axioms are objects, too - Ontology Engineering Beyond the Modeling of Concepts and Relations. In: V.R. Benjamins, A. Gomez-Perez, N. Guarino (eds.). *Proceedings of the ECAI 2000 Workshop on Ontologies and Problem-Solving Methods*. Berlin, August 21-22, 2000, pp. 8.1-8.6.
- [63] H.-P. Schnurr, ~. A Proactive Inferencing Agent for Desk Support. In: *Proceedings of the AAAI Symposium on Bringing Knowledge to Business Processes*. Stanford, CA, USA, March 20-22, 2000. AAAI Technical Report, Menlo Park, 2000, pp. 44-52.
- [64] A. Mädche, H.-P. Schnurr, ~, R. Studer. Representation Language-Neutral Modeling of Ontologies. In: Frank (ed.), *Proceedings of the German Workshop*

"Modellierung-2000". Koblenz, Germany, April, 5-7, 2000. Fölbach-Verlag, 2000, pp. 129-142.

- [65] M. Erdmann, A. Mädche, H.-P. Schnurr, ~. From Manual to Semi-automatic Semantic Annotation: About Ontology-based Text Annotation Tools. In: *P. Buitelaar & K. Hasida (eds). Proceedings of the COLING 2000 Workshop on Semantic Annotation and Intelligent Content*. Luxembourg, August 5-6, 2000.

1999

- [66] ~, T. Barnekow. Towards Learning Notification Triggers. In: *Proceedings of the international Workshop on Intelligent Workflow and Process Management: The New Frontier for AI in Business (IJCAI-99)*. Stockholm, Sweden, 1999.
- [67] ~, H.-P. Schnurr. Knowledge and Business Processes: Approaching an Integration. In: *OM '99 - Proceedings of the international Workshop on Knowledge Management and Organizational Memory (IJCAI-99)*. Stockholm, Sweden, 1999, pp. 140-159.
- [68] H.-P. Schnurr, ~, R. Studer. Ontology-based Process Support. *Workshop on Exploring Synergies of Knowledge Management and Case-Based Reasoning (AAAI-99)*. Technical Report, Menlo Park: AAAI.
- [69] ~, C. Braun, I. Bruder, A. Düsterhöft, A. Heuer, M. Klettke, G. Neumann, B. Prager, J. Pretzel, H.-P. Schnurr, R. Studer, H. Uszkoreit, & B. Wrenger. GETESS - Searching the Web Exploiting German Texts. *CIA '99 - Proceedings of the 3rd international Workshop on Cooperating Information Agents*. Upsala, Sweden, July 31-August 2, 1999, pp.113-124, LNAI 1652, Berlin, Heidelberg: Springer.
- [70] D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, ~, R. Studer, A. Witt. On2broker: Semantic-Based Access to Information Sources at the WWW. In: *Proceedings of the Workshop on Intelligent Information Integration (III99) during IJCAI-99*, Stockholm, Sweden, August 1999, pp. 63-67.
- [71] A. Mädche, ~, R. Studer. Ontology-oriented Information Extraction and Integration. Workshop on Language Technologies in Information and Knowledge Management. In: *Proceedings of the 7th Conference on Computational Linguistics of the German Society for Language Technologies*. Saarbrücken, Germany, October 7-8, 1999 (Abstract).

1997

- [72] ~, U. Hahn. A Semantic Copying Model for Understanding Comparatives. In: H. Bunt, L. Kievit, R. Muskens & M. Verlinden (eds.), *Proc. of the 2nd International Workshop on Computational Semantics, IWCS-II*, Tilburg/The Netherlands, J.uary 8-10, 1997, pp. 274-286, Department of Computational Linguistics, Tilburg University.
- [73] ~, U. Hahn. Dimensional Reasoning with Qualitative and Quantitative Distances. In: *AAAI-97 Workshop on Spatial and Temporal Reasoning*, Providence, RI, USA, July 27-31, 1997, Conference Technical Reports WS-97-11, pp. 63-72, Menlo Park/CA: AAAI Press.

Unveröffentlichte Technische Berichte

2002

- [74] 2002 R. Volz, ~, R. Studer. Views for semi-structured data in the Semantic Web. 2002.

2001

- [75] A. Mädche, ~. Comparing Ontologies – Similarity Measures and a Comparison Study. 6 Seiten (5600 Wörter). Technical Report 408, 2001.
- [76] H.-P. Schnurr, ~, R. Studer, Y. Sure. Ontologiebasiertes Wissensmanagement – Ein umfassender Ansatz zur Gestaltung des Knowledge Life Cycle. 2001.
- [77] M. Erdmann, A. Mädche, H. Oppermann, H.-P. Schnurr, ~, Y. Sure, R. Studer, C. Tempich. Arbeitsgerechte Bereitstellung von Wissen – Ontologien für das Wissensmanagement. 2001.

2000

- [78] I. Horrocks, D. Fensel, J. Broekstra, S. Decker, M. Erdmann, C. Goble, F. Van Harmelen, M. Klein, ~, R. Studer. *The Ontology Inference Layer OIL: The grease between Ontologies*. Technical report, 2000.
- [79] ~, A. Mädche, S. Handschuh. Creating Metadata for the Semantic Web: An Annotation Environment and the Human Factor. Technical report, 2000.

1999

- [80] A. Mädche, G. Neumann, ~. *A Generic Architectural Framework for Text Knowledge Acquisition*. Technical report, 1999.

1998

- [81] ~. *Grading Knowledge - Extracting Degree Information from Texts*. Ph.D. thesis (computer science), University of Freiburg, 1998.

1995

- [82] ~. *GLR-Parsing von Worthypothesengraphen*. Verbmobil Report, DFKI, Saarbrücken, NR. 85, 1995.

1994

- [83] ~. *GLR-Parsing of Word Lattices*. Master's thesis, University of Pennsylvania, 1994.

Posterbeiträge

2001

- [84] S. Handschuh, S. Staab, F. Ciravegna. S-CREAM – Semi-automatic CREation of Metadata. Poster Proceedings of ISCW-2002. Chia, Sardegna, Italy, June 2002.

2001

- [85] A. Hotho, A. Mädche, ~. Text Clustering Based on Good Aggregations. In: *Proc. of ICDM – IEEE International Conference on Data Mining 2001*, Vancouver, December 2001.
- [86] A. Mädche, ~: Ontology Learning from Text. In: M. Bouzeghoub, Z. Kedad, E. Métais (Eds.): *Natural Language Processing and Information Systems, 5th International Conference on Applications of Natural Language to Information Systems, NLDB 2000*, Versailles, France, June 28-30, 2000, pp. 364, LNCS 1959, Springer 2001.

1999

- [87] D. Fensel, A. Witt, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, ~, R. Studer. On2broker in a Nutshell. In: *WWW8 - Poster Proceedings of the 8th World Wide Web Conference*, Toronto, May 11-14, 1999.

Software Demonstration

2000

- [88] A. Mädche, ~: The Text-To-Onto Ontology Learning Environment. Software Demonstration at *ICCS-2000 - Eight International Conference on Conceptual Structures*. August, 14-18, 2000, Darmstadt, Germany.

Herausgeberschaften

In Vorbereitung

- [89] ~, R. Studer (eds.). *Handbook on Ontologies in Information Systems*. International Handbooks on Information Systems, Springer Verlag.
- [90] S. Handschuh, ~ (eds.). *Annotation for the Semantic Web*. IOS Press.

2002

- [91] M. Frank, N. Noy, ~ (eds.). *Semantic Web 2002 – Proceedings of the Workshop at WWW-2002*. Honolulu, Hawaii, May 7, CEUR Workshop proceedings, 2002.
- [92] ~, R. Studer (eds.). *Intelligente Systeme für das Wissensmanagement - KI - Zeitschrift Künstliche Intelligenz*. Schwerpunktheft 1/2002.
- [93] M. Minor, ~ (eds.). *Experience Management. Proceedings of the 1st Workshop of the German GI Interest Group on Knowledge Management*. Berlin, March 7-8, 2002. Lecture Notes in Informatics, P-10, Gesellschaft für Informatik e.V. / Köllen Verlag.

2001

- [94] S. Handschuh, R. Dieng, ~ (eds.). *Proceedings of the Workshop on "Semantic Annotation and Knowledge Markup"*. At K-Cap 2001. October 21-23, 2001, Victoria, B.C., Canada.

- [95] A. Mädche, ~, G. Stumme (eds.). *Proceedings of the Workshop on "Ontologies"*. At KI-2001, September 19-21, 2001 Wien, Österreich. In Vorbereitung.
- [96] ~, S. Decker, D. Fensel, A. Sheth (eds.). *Semantic Web 2001 – Proceedings of the Workshop at WWW-10*. Hongkong, China, May 1, 2001.
- [97] H.-P. Schnurr, ~, R. Studer, G. Stumme, Y. Sure (eds.). *WM2001 – Professionelles Wissensmanagement – Erfahrungen und Visionen. Proceedings der Konferenz WM2001*, Shaker Verlag, März 2001.
- [98] A. Mädche, ~, E. Hovy, C. Nedellec (eds.). *The IJCAI-2001 Workshop on Ontology Learning. Proceedings of the Second Workshop on Ontology Learning - OL'2001*, Seattle, WA, USA, August 4, 2001. CEUR Proceedings, Aachen. 2000
- [99] E. Tsui, B. Garner, ~. *AI in Knowledge Management – Special Issue of Elseviers "Knowledge-based Systems"*. Volume 13, Number 5, 2000.
- [100] ~, A. Mädche, C. Nedellec, P. Wiemer-Hastings (eds.). *The ECAI'2000 Workshop on Ontology Learning. Proceedings of the First Workshop on Ontology Learning - OL'2000*, Berlin, Germany, August 25, 2000. CEUR Proceedings, Vol-31.
- [101] ~, D. O'Leary (eds.). *"Bringing Knowledge to Business Processes. Papers from the 2000 AAAI Spring Symposium"*, Stanford, March 20-22, 2000, AAAI Technical Report SS-00-03, Menlo Park, CA.

Publikationen in Vorbereitung

Eingereicht zur Veröffentlichung:

- [102] C. Schmitz, ~ Staab, R. Studer, G. Stumme, J. Tane. Accessing Distributed Learning Repositories through a Courseware Watchdog. Submitted to *E-Learn-2002*.
- [103] Y. Sure, ~, J. Angele, D. Wenke, A. Mädche. OntoEdit: Guiding Ontology Developement by Methodology and Inferencing. Submitted to *ODBASE-2002*.
- [104] A. Mädche, V. Pekar, ~. Ontology Learning Part One – On Discovering Taxonomic Relations from the Web. Submitted to Ning Zhong et al. (eds) *Web Intelligence*. Springer.

Sonstige Veröffentlichungen

2002

- [105] ~, A. Mädche, F. Nack, S. Santini, L. Steels. Emergent Semantics. *IEEE Intelligent Systems, Trends & Controversies*, 17(1), Jan/Feb 2002, pp. 78 – 86.
- [106] ~, P. Clark, J. Hendler, I. Horrocks, P. Patel-Schneider, M.-C. Rousset, G. Schreiber, M. Uschold, F. van Harmelen. Ontologies' KISSES in Standardization. *IEEE Intelligent Systems, Trends & Controversies*, 17(2), Mar/Apr 2002, pp. 70-79.
- [107] ~, C. Blaschke, C. Nedellec, J. Park, B. Schatz, A. Valencia, Luca Bernardi, Esther Ratsch, Renate Kania, Jasmin Saric, and Isabel Rojas. Mining Information for Functional Genomics. *IEEE Intelligent Systems, Trends & Controversies*, 17(3), May/June 2002.

- [109] ~. Knowledge Portals. *Künstliche Intelligenz*. Stichwort – Themenheft Wissensmanagement. 2002, 1.

2001

- [110] ~. Semantic Web – Das Web der nächsten Generation. *Karlsruher Transfer*, 15, Sommersemester 2001, pp. 18-23.
- [111] A. Mädche, ~, R. Studer. Ontologien. In *Wirtschaftsinformatik* (WI-Schlagwort), 43(4): 393-396, 2001.
- [112] ~, S. Handschuh, A. Mädche. Metadata and the Semantic Web – and CREAM (Extended Abstract of Invited Talk). In *Proceedings of the DELOS-2001 workshop*. September 8-9, 2001, Darmstadt, ECRIM, 2001.
- [113] ~. Learning and the Semantic Web (Extended Abstract of Invited Talk). In *Proceedings of Benelearn 2001 – The Annual Machine Learning Conference of Belgium and The Netherlands*, Antwerp, Belgium, December 21, 2001.
- [114] ~, F. Ciravegna, M. Maybury, D. Moldovan, G. Niklfeld. Human language technologies for knowledge management. *IEEE Intelligent Systems*, Trends & Controversies, 16(6), Nov/Dec 2001, pp. 84 –94.